UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

GRADO EN INGENIERÍA INFORMÁTICA

Amenazas y defensa en sistemas ciber-físicos con conexión al mundo virtual:
Detección de anomalías en sistemas CPS mediante machine learning

Threats and defense in cyber-physical systems with connection to the virtual world:
Anomaly-based detection in CPS systems through machine-learning

Realizado por
Marta Ferrer Cuesta

Tutorizado por
María Cristina Alcaraz Tello
Francisco Javier López Muñoz

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, Septiembre de 2020

Resumen

Los sistemas ciberfísicos se caracterizan por tener capacidad de comunicación y computación pudiendo trabajar en sistemas distribuidos y de forma autónoma. Están presentes en muchas actividades de la vida cotidiana, desde las industrias y los hospitales, hasta los hogares, e incluso, el cuerpo humano. La interconexión conlleva riesgos y amenazas en la seguridad de los datos y las comunicaciones que fuerzan al desarrollo de nuevos mecanismos de detección de anomalías y prevención de ciberataques.

Este TFG, por tanto, se enmarca en la protección de infraestructuras críticas y, concretamente, en el ámbito de la salud donde la seguridad y el correcto desarrollo de los procesos que tienen lugar llega a ser fundamental. El sistema desarrollado tiene el objetivo de proporcionar una monitorización en tiempo real de una sala de operaciones y la detección de posibles anomalías que puedan ocurrir en ella a través de una visualización en 3D. Para la detección de anomalías se han empleado técnicas de *machine-learning* a datos obtenidos de sensores conectados a un microcontrolador y para la visualización se ha tratado de diseñar una interfaz amigable y realista mediante el modelado 3D en base a una entrevista a un cliente real y documentos oficiales.

En el futuro se irán introduciendo cada vez más aplicaciones de IoT, análisis de datos y realidad virtual y aumentada para proporcionar asistencia a los médicos con el fin de hacer su trabajo más preciso y eficiente. Este proyecto es sólo una pincelada de las posibilidades de un campo en el que aún queda mucho por investigar y que cambiará la medicina tal y como se conoce hoy en día.

Palabras clave

sistemas ciber-físicos, machine-learning, detección de anomalías, detección de intrusiones

Abstract

Nowadays cyber-physical systems are present in most domains of everyday life, from industry to hospitals and even the human body. Cyber-physical systems are those which integrate physical and computation algorithms and are capable of operating in distributed systems. Connected systems and wireless communications imply risks and hazards for data and communications security which results in the challenge of developing solutions so as to detect anomalies and protect.

This project focuses on critical infrastructures, particularly in relation to health, in which security and safe development of processes that take place in are critical. The main goal of the developed system is monitoring an operation theatre in real-time detecting anomalous behaviours with a 3D visualization. Machine-learning techniques have been applied to detect deviations in data collected from sensors as well as intrusion detection techniques such as whitelists. A 3D environment has been designed for visualisation. The design of the system and the kind of sensors used have been based in an interview and official documents.

In the future, more and more IoT, data analysis, virtual and augmented reality applications will be developed in order to assist doctors so as to do a better and more accurate work. Therefore, this work is part of a challenging field that will change healthcare although more research is needed.

Keywords

Cyber-physical systems, machine-learning, anomaly detection, intrusion detection

Índice

Índice de tablas

Índice de figuras

Lista de Acrónimos

- **BSON** Binary JavaScript Object Notation
- **CPS** Cyber Physical System
- **DE.AE** Detection Anomalies and Events
- **DE.CM** Detection Continuous Monitoring
- **DE.DP** Detection Detection Process
- **RS.CO** Response Comunications
- **ECM** Error Cuadrático Medio
- **GPU** Licencia pública general de GNU
- **HIPAA** Health Insurance Portability and Accountability Act
- **TIC** Tecnologías de la Información y la Comunicación
- **IDE** Integrated Development Environment
- **ID.AM** Identification Active Management
- **ID.BE** Identification Business Environment
- **ID.RA** Identification Risk Assessment
- **IoT** Internet of Things
- **I2C** Inter Integrated Circuits
- **JSON** Javascript Object Notation
- **LAN** Local Area Network
- **LGPL** Licencia pública general reducida de GNU

- **NIST** National Institute of Standards and Technology
- **NSF** National Society Foundation
- **PR.AC** Protection Access Control
- **PR.DS** Protection Data Security
- **RAM** Random Access Memory
- **RRNN** Redes Neuronales
- **RS.CO** Response Communications
- **SCA** System Data
- **SCL** System Clock
- **SVM** Support Vector Machine
- **WAN** Wide Area Network
- **3D** Tres dimensiones

1. Introducción

A lo largo de este apartado se expone la motivación que ha llevado a la elaboración de este proyecto, el estado actual de la cuestión, los objetivos que se pretenden abarcar y la organización que sigue el presente documento.

1.1. Motivación

Actualmente nos encontramos ante una revolución tecnológica en la que la inteligencia artificial, los sistemas ciberfísicos, y el IoT (Internet of Things), entre otros, desempeñan un papel fundamental en la sociedad. La NSF (National Society Foundation), define los sistemas ciberfísicos (o Cyber Physical Systems (CPS), en inglés) como sistemas que integran componentes físicos y algoritmos computacionales [1]. Estos sistemas deben operar de forma dependiente, segura, eficiente y en tiempo real.

Los sistemas CPS proporcionan la capacidad de monitorizar y controlar la dinámica continua de los sistemas físicos, sus aplicaciones abarcan cualquier ámbito del mundo físico como son la automatización de fábricas, la construcción de espacios inteligentes o la salud. Muchas de estas aplicaciones son infraestructuras críticas, las cuales precisan de un control exhaustivo de los procesos que tienen lugar en ellas [2].

La digitalización del entorno se encuentra sujeta a la exposición a vulnerabilidades de seguridad en las comunicaciones que pueden provenir de diversas fuentes [3]: un mal uso del sistema, ataques externos a través de la red o incluso por el fallo de algún componente, ya sea éste hardware o software [4].

Este proyecto se enmarca en el ámbito de la salud, concretamente en la denominada Salud 4.0., concepto surgido del aporte de las TIC (Tecnologías de la Información y la Comunicación) dentro de este sector. Entre los beneficios de la aplicación de CPS al ámbito de la salud se encuentran la reducción de costes y una mayor efectividad que repercute en

el beneficio de los pacientes gracias a la simulación y la personalización de los tratamientos empleando técnicas como la realidad virtual y el procesamiento de imágenes. Por lo tanto, la motivación de este proyecto reside en la protección de las infraestructuras críticas sanitarias ante las vulnerabilidades de seguridad de las comunicaciones y la detección de anomalías en los datos proporcionados por los sensores integrados en los sistemas de CPS.

Para conocer mejor la situación actual en los hospitales se contó con la oportunidad de concertar una entrevista con un doctor del hospital de alta resolución de Benalmádena, tras la cual se decidió centrar la aplicación en una sala de operaciones. En una sala de operaciones, además de las constantes vitales del paciente, deben tenerse en cuenta parámetros ambientales controlados por sensores. Otro aspecto a tener en cuenta es que los robots perciben el mundo gracias a los sensores que llevan incorporados, por todo esto es indispensable el análisis de los datos recogidos por los sensores ya que cada vez hay en el mercado más robots médicos con mayores capacidades e interconexión. Además, el avance tecnológico ha propiciado la aparición de herramientas que permiten desarrollar sistemas de realidad aumentada o virtual para simular y monitorizar situaciones con interfaces amigables.

1.2. Estado del arte

El área de la Salud 4.0. es un ámbito de aplicación de la Industria 4.0., cuyos principios de diseño son: interoperabilidad, virtualización, descentralización, tiempo real y modularidad. Uno de los desafíos que plantea la Industria 4.0 es la seguridad, pues la interconexión implica una mayor apertura y con ello, el riesgo de ciberataques. Otro reto es la fiabilidad de estos sistemas, para ello se están desarrollando soluciones de análisis dirigidas por datos (*data-driven*) para el diagnóstico, y la anticipación a los fallos, errores o anomalías con el objetivo de evitarlos o reducir su impacto [5].

La supervisión de un proceso consiste en una serie de acciones y tareas con el propósito de asegurar un correcto funcionamiento de los sistemas y procesos, incluso en situaciones

anómalas. La supervisión se realiza a través de encargados y operarios especializados, en el caso de una sala de operaciones el personal sanitario, que detectan la presencia de comportamientos anómalos y actúan en consecuencia. Para ello, se debe poder registrar la evolución del proceso y detectar desviaciones y anomalías, para posteriormente analizar esas desviaciones, y elaborar un diagnóstico de la situación. Si el sistema de supervisión no contiene las etapas de detección, diagnóstico y reconfiguración; se trata de un sistema de monitorización o de vigilancia de ayuda al usuario en la toma de decisiones: sistemas que alertan al operario para que éste decida sobre la existencia de fallos, su origen y los pasos a seguir en ese caso [6]. En las infraestructuras críticas con presencia de sistemas ciberfísicos, la ciberseguridad y la detección de anomalías son, por tanto, áreas fundamentales para la protección.

Los sistemas ciberfísicos son sistemas que integran componentes físicos como son sensores y actuadores con algoritmos computacionales, e incluso, interconexión inalámbrica. Estos sistemas dotan de inteligencia y conectividad a elementos y procesos cotidianos. En la figura 1 podemos ver varios ámbitos de aplicación de estos sistemas.

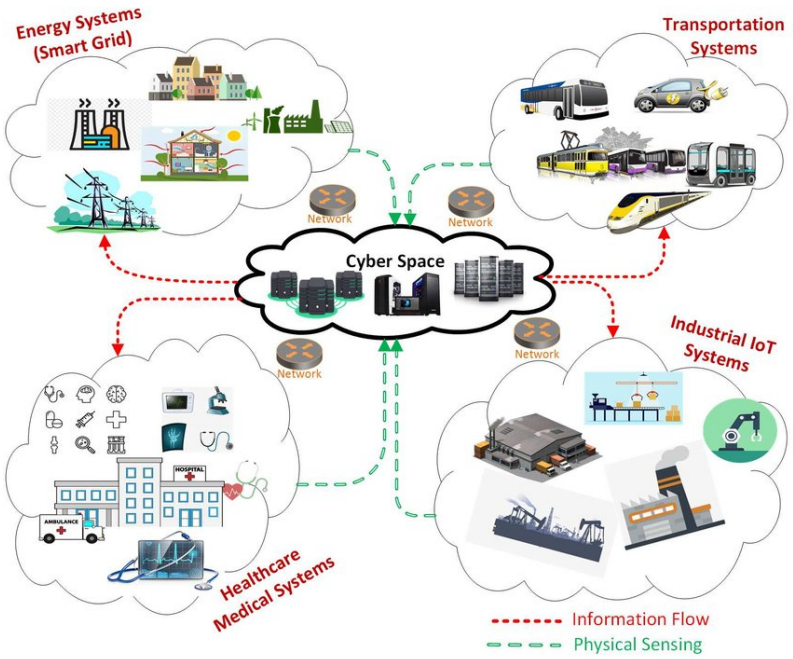


Figura 1: Escenarios de aplicación de CPS [7]

Los sistemas ciberfísicos del ámbito de la salud incluyen wearables, instrumental de las salas de operaciones y elementos hardware fisiológicos (por ejemplo, marcapasos). En este proyecto se ha puesto el foco en las salas de operaciones y su instrumental, aunque su aplicación puede ser extrapolable a otras aplicaciones de sistemas CPS.

Gracias al avance tecnológico y la mejora de las telecomunicaciones ha surgido la medicina digital y robotizada que permite obtener resultados y niveles de precisión sin precedentes [8]. Ya en 2001 se operó a distancia a un hombre en Francia con un robot dirigido por cirujanos en Nueva York, y hoy en día se realizan operaciones a distancia gracias al avance tecnológico [9]. Estas actividades son críticas desde el punto de vista de la seguridad, pues el correcto desarrollo de los procesos que tienen lugar repercute directamente en la salud y el bienestar de las personas.

Detección de anomalías

Este TFG se centra en la protección de infraestructuras críticas mediante el análisis de datos, concretamente orientado a la detección de anomalías. Para la detección de anomalías hay diversas técnicas en la literatura [10]:

- **Métodos estadísticos**: empleo de modelos estadísticos como modelos de Markov, análisis paramétrico o técnicas basadas en series temporales.
- ***Machine-Learning***: encontrar patrones en el conjunto de datos o muestras.

El *Machine-learning* (aprendizaje automático) surge debido a la necesidad de extraer información útil de los datos, y forma parte del campo de la Inteligencia artificial. Esta técnica es la más adecuada cuando se desconocen las relaciones entre los datos. Dentro del *machine-learning*, los algoritmos pueden clasificarse en función de cómo se realiza el entrenamiento de los modelos en: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje semi-supervisado [11, 12].

- **Aprendizaje supervisado**: los datos de entrenamiento se encuentran clasificados. Este tipo de aprendizaje trata de encontrar el modelo que más se aproxime a el valor de clasificación según los datos de entrada. Este método es el que mayor efectividad tiene, sin embargo, tener grandes conjuntos de datos etiquetados requiere un gran esfuerzo. Además, en casos como la detección de anomalías no sólo se requieren grandes cantidades de datos, sino que la mayor parte de los datos son normales habiendo una minoritaria cantidad de anomalías quedando un conjunto de datos descompensado. Los algoritmos supervisados más comunes en detección de anomalías son: RRNN (Redes Neuronales), *SVM* (Support Vector Machine).

- **Aprendizaje no supervisado**: en este tipo de entrenamiento los datos no se encuentran clasificados. En este caso no se trata de clasificar cada observación (conjunto de datos que constituyen una entrada al modelo), sino de identificar patrones e información a partir de los datos. Algoritmos: *k-NN*, *K-means*, *Autoencoders*, entre otros.

- **Aprendizaje semi-supervisado**: también denominado aprendizaje por refuerzo. En este método de aprendizaje los algoritmos tienen el objetivo de tomar la decisión que maximice la recompensa. Por ejemplo: el algoritmo *Q-Learning*.

1.3. Objetivos del TFG

El objetivo principal de este proyecto, como ha sido expuesto en apartados anteriores, se enmarca en el ámbito de las infraestructuras críticas y, más concretamente, en el de la Salud 4.0. Al tratarse de un proyecto grupal contiene partes comunes y partes individuales.

En este TFG se ha desarrollado un sistema de CPS para monitorizar lo que ocurre en una sala quirúrgica mediante una representación gráfica en 3D y la presencia de sensores que captan información del entorno en tiempo real. A su vez, los datos recogidos se emplean para detectar e informar de fallos y situaciones anómalas.

Los tres objetivos principales de desarrollo son:

1. Modelado de un escenario del mundo real en 3D.(Objetivo grupal).
2. Ataques al sistema CPS desarrollado. (Realizado en el TFG "Amenazas y defensa en sistemas ciber-físicos con conexión al mundo virtual: Modelos de amenazas específicos para sistemas CPS" de Pablo Gutiérrez Ruiz).
3. Defensa mediante el uso de *machine learning* para la derección de anomalías e intrusiones. (Desarrollado en este TFG).

En primer lugar, el modelado consiste en una representación gráfica en 3D interactiva de una sala quirúrgica que reflejará la información recogida del escenario de aplicación mediante distintos tipos de sensores en tiempo real.

La naturaleza crítica del entorno requiere que la información obtenida de los sensores y de los eventos que tengan lugar queden reflejados en una base de datos no relacional. Esta elección se justifica en la escalabilidad que proporciona para incluir nuevos sensores y que no es necesario conocer las relaciones entre los datos para que sean almacenados. Todo esto formará parte del núcleo de los dos TFGs en los que se desarrolla este sistema.

Gracias al almacenamiento de esta información es posible entrenar modelos de detección de anomalías y posibles ataques al sistema y sus comunicaciones con técnicas de machine learning. Es en este ámbito de desarrollo en el que se centra este TFG.

El modelado de los ataques al sistema desarrollado y a las comunicaciones se describe en el TFG "Amenazas y defensa en sistemas ciber-físicos con conexión al mundo virtual: Modelos de amenazas específicos para sistemas CPS".

El mecanismo de detección consistirá en un algoritmo de *machine-learning* que detectará las situaciones anómalas que puedan estar provocadas por un funcionamiento inadecuado del sistema o por algún tipo de ataque, y que como consecuencia puedan

suponer un peligro para la integridad física tanto de los pacientes como del personal sanitario presente en una sala de operaciones. Dichos algoritmos deberán analizar y obtener respuestas en tiempo real, ya que así lo exige el escenario de trabajo.

1.4. Organización de la memoria

La organización de este TFG es la siguiente:

- **Capítulo 1 - Introducción**: se expone la motivación que ha llevado a la elaboración de este proyecto, un resumen sobre el contexto y el estado del arte en el que se enmarca, y los objetivos que se pretenden alcanzar.
- **Capítulo 2 - Metodología**: se aborda la organización que se ha llevado a cabo para el desarrollo de este proyecto, indicando de forma muy resumida las fases de elaboración del mismo y su cronología.
- **Capítulo 3 - Digitalización del mundo real**: se muestra un análisis de los requisitos que debe cumplir el sistema siguiendo las directrices elaboradas por organismos oficiales de estandarización y el caso de uso implementado.
- **Capítulo 4 - Tecnologías hardware y software empleadas**: este capítulo contiene una breve introducción de las tecnologías hardware y software empleadas en el desarrollo del sistema así como el motivo por el cual han sido seleccionadas.
- **Capítulo 5 - Arquitectura del sistema**: en este capítulo se añade la estructura del sistema, explicando cómo ha sido implementado y cómo se relacionan las partes que lo componen.
- **Capítulo 6 - Detección de anomalías e intrusiones**: a lo largo del capítulo se realiza una explicación detallada de los algoritmos de *machine-learning* encargados de detectar tanto anomalías en los datos, como intrusiones. También se muestra un análisis de su funcionamiento.

- **Capítulo 7 - Conclusiones y trabajo futuro**: en este último capítulo se incluyen las conclusiones extraídas del análisis de los objetivos alcanzados y cuáles han sido las dificultades encontradas. También se incluyen propuestas y sugerencias para un trabajo futuro.

Aparte de esto, este TFG cuenta con tres anexos con información detallada y complementaria sobre el sistema desarrollado: dos manuales; uno de usuario, con los pasos a seguir para la puesta en marcha del sistema, y el otro, de instalación. Además de esto, se añade la entrevista realizada a un médico del Hospital de Alta Resolución de Benalmádena.

2. Metodología

Para la elaboración de este proyecto se ha empleado la metodología Scrum [13] puesto que se trata de una metodología ágil de desarrollo incremental con iteraciones cortas. En este caso, el desarrollo se ha organizado en sprints cada dos semanas, planificando al final de cada uno de ellos una reunión con los tutores (Scrum Masters) en la que se exponía un resumen de los avances, las dificultades encontradas y se realizaba la planificación del siguiente sprint.

Como herramienta complementaria se ha utilizado Trello (figura 2) [14], que aunque está pensada para seguir la metodología Kanban [15] ayuda a llevar un control visual de las tareas pendientes, en proceso y las ejecutadas.

Figura 2: Captura Trello

Los sprints se han dividido según las distintas fases de desarrollo:

- **Estudio para la digitalización del mundo real** (grupal): durante esta fase se han establecido los requisitos del sistema, y se ha definido el caso de uso. También se han formulado una serie de preguntas con el fin de entrevistar a un médico. Una vez realizada la entrevista, se han establecido algunos parámetros que necesariamente deben ser controlados en el curso de una intervención, y cuál es la normativa que

debe cumplirse.

- **Comunicación con los sensores y Arduino** (grupal): la siguiente fase de desarrollo ha consistido en la elección de los sensores así como la familiarización con la programación del microcontrolador Arduino.
- **Diseño y modelado del escenario** (grupal): en esta etapa de desarrollo se ha modelado una sala de operaciones en 3D.
- **Creación de la base de datos y las comunicaciones** (grupal): se ha creado una base de datos no relacional conectada con el sistema mediante el lenguaje de programación Python, el cual es empleado para guardar los datos de los sensores y el análisis de anomalías. Tambien se ha conectado a la base de datos el escenario virtual mediante el lenguaje de programación C#.
- **Estudio e implementación de los algoritmos de *machine-learning*** (individual): se han valorado qué algoritmos son los más apropiados para detectar las anomalías y finalmente se ha implementado un algoritmo basado en redes neuronales.
- **Pruebas y experimentación** (grupal): se ha puesto a prueba el sistema implementado.

En la figura 3 tenemos un diagrama de Gantt que es una representación visual del desarrollo.

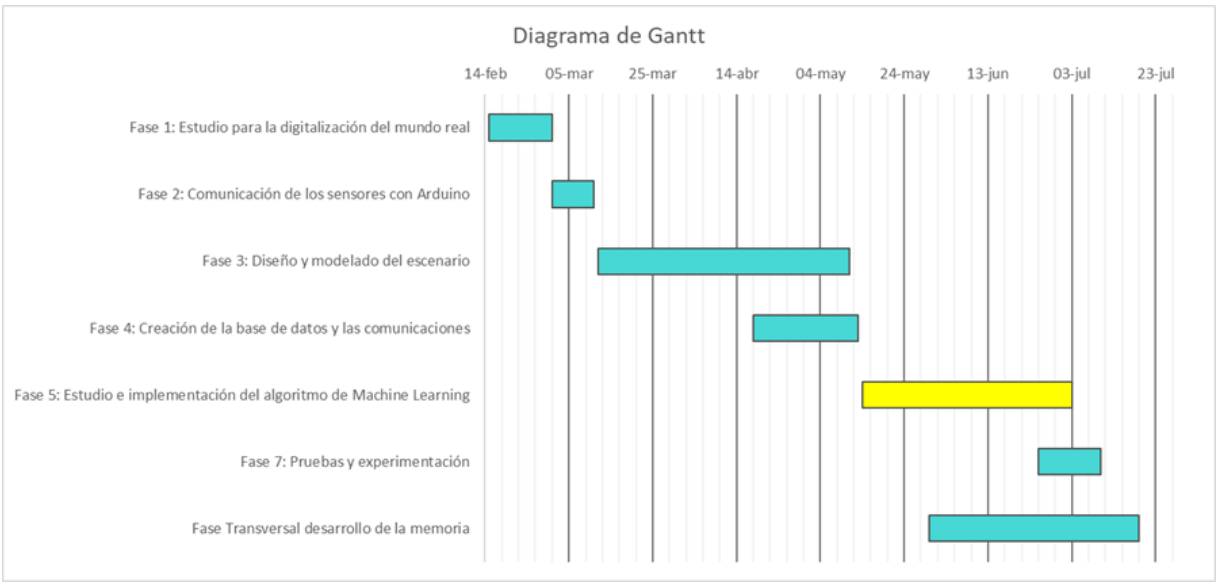


Figura 3: Diagrama de Gantt

3. Digitalización del mundo real

A lo largo de este apartado se expondrán los requisitos establecidos para el sistema, así como los casos de uso. Ambos fueron determinados tras mantener una entrevista con un cliente (anexo C), un médico del Hospital de Alta Resolución de Benalmádena, y varias reuniones con los tutores.

3.1. Requisitos del sistema

La elaboración de los requisitos para el sistema que se presenta se ha basado en el Marco para la mejora de la seguridad cibernética en infraestructuras críticas [16] y las reglas HIPAA (Health Insurance Portability and Accountability Act) [17].

Esta reglamentación no es de obligado cumplimiento, pero contiene unas directrices de ciberseguridad en infraestructuras críticas basadas en la experiencia y el avance tecnológico que han sido aprobadas por un organismo de normalización reconocido como es el NIST (National Institute of Standards and Technology) y constituyen un consenso entre todos los expertos de las materias que intervienen en el desarrollo de las actividades relacionadas con la Industria 4.0.

A continuación, se describen las reglas HIPAA y las actividades del marco para la mejora de la seguridad cibernética en infraestructuras críticas empleadas.

Las reglas HIPAA, se trata de una ley estadounidense [18] en la cual se establecen estándares para garantizar la seguridad y privacidad de los pacientes y sus datos con el objetivo de mejorar la eficiencia y efectividad de los sistemas sanitarios.

Por su parte, el Marco de ciberseguridad es un documento elaborado por el NIST para la evaluación y gestión del riesgo cibernético de las infraestructuras críticas. En él se establecen principios y buenas prácticas para la gestión de la seguridad cibernética. Las funciones del marco de ciberseguridad que se abordan en este proyecto son las de:

Identificar (ID), Proteger (PR), Detectar (DE) y Responder (RS) ante riesgos y amenazas.

Los requsitos para el sistema CPS desarrollado aparecen detallados en tablas que se presentan a continuación estructuradas según las funciones que se mencionan en el párrafo anterior. Dentro de cada función hay varios requisitos del marco de ciberseguridad que se aplican al sistema, y cada requisito puede estar asociado con una o varias reglas HIPAA.

En la función de identificación (ver tabla 1) se abordan: la gestión de activos, el entorno empresarial y la evaluación de los riesgos. En la protección (ver tabla 2): el control de acceso y la seguridad de los datos. En la detección (ver tabla 3): la detección de las anomalías y eventos, el monitoreo continuo de la seguridad y los procesos de detección. Por último, en la función de respuesta (ver tabla 4): la comunicación entre los usuarios y el sistema.

GESTIÓN DE ACTIVOS (AM)			
ID.AM-6	Estarán descritos todos los roles que actúan en el sistema.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: general rules. (a.3) (protección ante usos y revelación de información). Regla HIPAA - § 164.308 Administrative safeguards (a.3.i).		
Afecta a	Sistema CPS (base de datos)	Administrador	Personal Sanitario
Precondición	Hay distintos usuarios, dispositivos y procesos que deben tener asignados permisos y privilegios adecuados.		
Postcondición	Los usuarios deberán tener asignados los roles correspondientes.		
ENTORNO EMPRESARIAL (BE)			

ID.BE-2	Se establecerá el escenario en el que se sitúa el sistema desarrollado y los parámetros que se monitorizarán.		
Afecta a	Sistema CPS		
Precondición	El objetivo del sistema será establecido.		
Postcondición	Quedará establecido el entorno de actuación del sistema, en este caso una sala quirúrgica.		
EVALUACIÓN DE RIESGOS (RA)			
ID.RA-3	Las vulnerabilidades a las que Sistema CPS tiene que hacer frente serán identificadas.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.2)*(Protección de la integridad de los datos).		
Afecta a	Sistema CPS		
Precondición	Al tratarse de una infraestructura crítica habrá que analizar todos los posibles riesgos y amenazas a los que el Sistema CPS puede estar expuesto.		
Postcondición	Las medidas para prevenir las vulnerabilidades serán implementadas.		

Tabla 1: Requisitos de identificación

CONTROL DE ACCESO (AC)	
PR.AC-1	Los usuarios y contraseñas se gestionarán para dispositivos, usuarios y procesos autorizados (estos usuarios y contraseñas estarán gestionados por MongoDB).
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3)(Protección ante usos y revelación de información) Regla HIPAA - § 164.308 Administrative safeguards. (a.3.i)

Afecta a	Sistema CPS	Personal sanitario y técnico	
Precondición	Identificar qué dispositivos, usuarios y procesos tienen que tener acceso a la base de datos.		
Postcondición	Cada dispositivo, usuario y proceso tendrá un usuario y contraseña.		
PR.AC-2	En el momento en el que el sistema esté implementado en un hospital real, el acceso a los dispositivos y sensores deberá estar restringido.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3) (Protección ante usos y revelación de información)		
Afecta a	Hospital	Sistema CPS	
Precondición	Establecer las medidas de restricción de acceso a dispositivos y sensores.		
Postcondición	Los dispositivos y sensores contarán con las medidas de protección adecuadas para acceder a ellos y la información que contienen.		
PR.AC-4	Los accesos a la base de datos garantizarán el principio de mínimos privilegios.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3) (Protección ante usos y revelación de información) Regla HIPAA - § 164.308 Administrative safeguards. (a.3.i)		
Afecta a	Sistema CPS (Base de datos)	Personal sanitario y técnico	
Precondición	Establecer los usos de la base de datos de cada usuario.		

Postcondición	Los usuarios contarán con aquellos privilegios que le permitan acceder exclusivamente a aquellos recursos a los que están autorizados.		
PR.AC-6	El sistema Sistema CPS verificará todas las credenciales de seguridad introducidas en el sistema.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3) (Protección ante usos y revelación de información)		
Afecta a	Sistema CPS	Personal sanitario y técnico	
Precondición	Un usuario de Sistema CPS ha sido dado de alta, tendrá un nombre de usuario y una contraseña.		
Postcondición	El usuario será autenticado correctamente.		
SEGURIDAD DE LOS DATOS (DS)			
PR.DS-2	Las comunicaciones de los datos estarán protegidas.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.1) (a.2) (Protección de la integridad de los datos)		
Afecta a	Sistema CPS		
Precondición	Se habrá obtenido un certificado SSL para establecer comunicaciones seguras entre los sensores y la base de datos.		
Postcondición	La comunicación de los datos queda protegida..		

Tabla 2: Requisitos de protección

ANOMALÍAS Y EVENTOS (AE)

DE.AE-3	Los datos de sensores y de los paquetes TCP/IP entre los sensores y la base de datos serán recopilados y almacenados en la base de datos no relacional.		
Afecta a	Sistema CPS		
Precondición	Existe un flujo de datos de sensores y una comunicación con la base de datos a través de la red.		
Postcondición	La información queda almacenada de forma estructurada para su posterior utilización.		
DE.AE-5	Se analizarán posibles anomalías y se establecerán umbrales de alerta mediante algoritmos de *machine-learning*.		
Afecta a	Sistema CPS		
Precondición	Han sido establecidas las condiciones normales del escenario y los algoritmos de machine learning han sido entrenados.		
Postcondición	Las anomalías deberán ser detectadas correctamente.		
MONITOREO CONTINUO DE LA SEGURIDAD (CM)			
DE.CM-1	La red será monitorizada para detectar posibles ataques.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.2) (a.3) (Protección de la integridad y, el uso y revelación de información)		
Afecta a	Sistema CPS		
Precondición	Se producen comunicaciones a través de la red.		
Postcondición	El tráfico es analizado para detectar anomalías.		
DE.CM-2	La sala quirúrgica será monitorizada mediante una representación adecuada.		
Afecta a	Sistema CPS	Sala quirúrgica	
Precondición	Los sensores recogen datos del entorno físico.		

Postcondición	Se pueden observar los cambios en el entorno físico en tiempo real.		
PROCESOS DE DETECCIÓN (DP)			
DE.DP-3	Los procesos de detección serán probados.		
Afecta a	Sistema CPS		
Precondición	Se establecen los procesos necesarios de detección.		
Postcondición	Los procesos están verificados para su utilización.		
DE.DP-4	Se especificará el origen de la anomalía.		
Se asocia a	Regla HIPAA - § 164.404 Notification to individuals		
Afecta a	Sistema CPS	Personal sanitario	
Precondición	Se ha producido una anomalía.		
Postcondición	Se conoce de qué elemento de Sistema CPS procede la situación anómala.		
DE.DP-5	Los algoritmos de detección serán mejorados periódicamente.		
Afecta a	Sistema CPS		
Precondición	Se establece la frecuencia de mejora de los algoritmos de detección.		
Postcondición	Los algoritmos quedan actualizados.		

Tabla 3: Requisitos de detección

COMUNICACIONES (CO)	
RS.CO-1	Los responsables deberán conocer sus roles a la hora de tomar una decisión ante una incidencia.
Se asocia a	Regla HIPAA - § 164.308 Administrative safeguards. (a.5. i)

Afecta a	Sistema CPS	Personal sanitario	
Precondición	Se estudian las responsabilidades de cada usuario.		
Postcondición	El sistema es utilizado correctamente.		
RS.CO-2	Las anomalías serán notificadas mediante la pantalla de la simulación.		
Afecta a	Sistema CPS	Personal sanitario	
Precondición	Se detecta una anomalía.		
Postcondición	El personal queda informado de la anomalía y actuará en consecuencia.		

Tabla 4: Requisitos de respuesta

3.2. Caso de uso

El sistema desarrollado en este TFG proporciona los mecanismos necesarios para la supervisión de la temperatura y humedad, así como de la posición de la mesa de operaciones en una sala quirúrgica durante el desarrollo de una intervención, detectando situaciones anormales.

Los usuarios que intervienen en la supervisión de la intervención serían tanto los miembros del personal sanitario como el personal técnico. Los procesos que intervienen en el sistema como lo son el modelo de detección de anomalías, la placa Arduino que recoge los datos de los sensores, el programa que captura el tráfico de red y la base de datos también son actores. Todos los actores que interactúan con el sistema son:

- **Administrador/técnico**: persona encargada del mantenimiento del sistema de monitorización.
- **Personal sanitario**: miembros del personal sanitario que monitorizan la sala quirúrgi-

ca y que harán uso del sistema.

- **Arduino**: placa que envía los datos recogidos por los sensores.
- **Sniffer**: programa que recoge el tráfico de red dirigido a la base de datos.
- **Base de datos**: sistema de base de datos que almacena y gestiona los datos de tráfico de red y sensores.

Los casos de uso describen la interacción de los actores con el sistema que refleja el comportamiento que tendrá el sistema ante una determinada acción de cualquiera de los actores. Los diagramas de caso de uso son la representación gráfica en la que aparecen relacionados los actores y los componentes del sistema sistema. En la figura 4 podemos ver el diagrama obtenido para el sistema de CPS desarrollado.

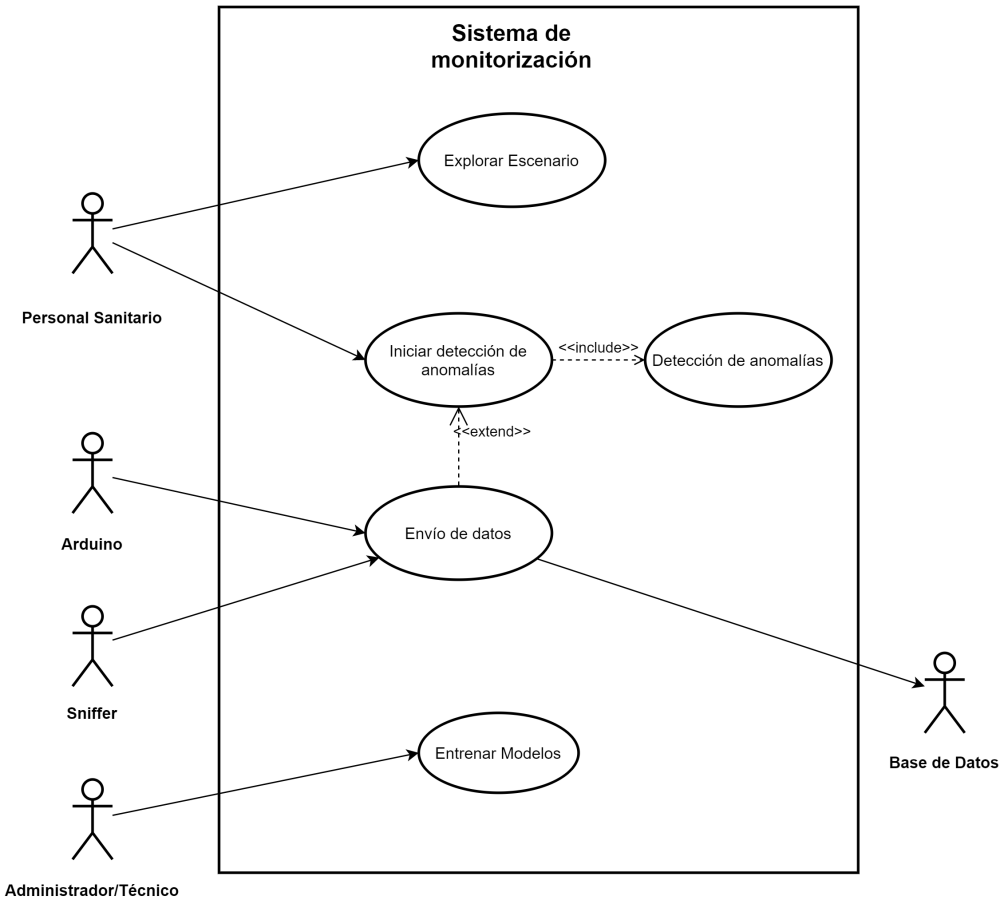


Figura 4: Diagrama de Casos de Uso

4. Tecnologías hardware y software empleadas

Las tecnologías utilizadas y los motivos por los que han sido seleccionadas para este proyecto se detallan en este apartado. Para la recogida de datos del escenario ha sido elegida la plataforma de desarrollo de Arduino junto a sensores de posición y de temperatura y humedad, como motor del escenario 3D la herramienta seleccionada ha sido Unity junto al lenguaje de programación C#, MongoDB como gestor de datos y Python como lenguaje para el desarrollo de los algoritmos de Machine Learning.

4.1. Hardware

4.1.1. Microcontrolador Arduino

Arduino [19] es una plataforma de desarrollo de hardware y software libre bajo las licencias GPU y LGPL. Esta plataforma permite el desarrollo rápido y sencillo de programas para controlar distintos sensores y actuadores. La programación y comunicación con estos microcontroladores es posible gracias a la comunicación serie con el ordenador. Estas aracterísticas junto a la flexibilidad que aporta a la hora de añadir periféricos compatibles con Arduino y ampliar con ellos sus capacidades hacen que sea una muy buena opción para este proyecto.

Hay varios tipos de placas Arduino, en este proyecto se han empleado dos:

- **Arduino UNO**: esta placa cuenta con un microcontrolador ATMega328P. Tiene 14 pines de E/S digitales, de los cuales 6 pueden ser utilizados como analógicos si es necesario y 6 pines analógicos.
- **Arduino MEGA**: en esta otra el microcontrolador es un ATMega2560 y cuenta con 54 pines de E/S digitales, de los que 15 pueden ser utilizados como PWM, y 16 pines analógicos.

El motivo de emplear dos placas reside en la cantidad de sensores de cada tipo requeridos para garantizar la seguridad en esta infraestructura crítica, al menos tres de cada tipo. Para asegurar que se está produciendo una anomalía real el sistema debe ser capaz de distinguir si sólo un sensor está presentando valores anómalos o bien son varios los sensores que están detectando valores que no son aceptables para el escenario.

4.1.2. Sensores

Sonar HC-SR04

Este sensor ha sido escogido para recoger datos de posición de la mesa de operaciones en una sala quirúrgica. El hecho de que la mesa de operaciones no sufra cambios de posición es muy importante, ya que el movimiento de la misma podría suponer poner una pérdida de precisión en la intervención y tener consecuencias fatales en la salud del paciente.

Figura 5: Sensor de ultrasonido HC-SR04

Con el sensor de ultrasonido HC-SR04 (ver figura 5) [20] se puede medir la distacia a la que se encuentra un objeto en un rango de trabajo de 2cm a 400cm y para un ángulo de 15 grados. Este sensor tiene un emisor y un receptor de ultrasonido, se emite ultrasonido durante un período de tiempo y posteriormente el receptor capta el ultrasonido reflejado al colisionar con el primer objeto encontrado en su recorrido. El sensor devuelve el tiempo

transcurrido desde que el ultrasonido en sale y vuelve al sensor. Para transformar este tiempo en centímetros se emplea la siguiente fórmula basada en la velocidad del sonido (343 m/s):

$$distancia(cm) = \frac{duracion(\mu s)}{2} * \frac{1(s)}{1000000(\mu s)} * \frac{343(m)}{1(s)} * \frac{100(cm)}{1(m)} \quad (1)$$

Cabe destacar, que este sensor ha sido escogido dado su bajo coste. Sin embargo, si el sistema fuese implantado en un escenario real es importante saber actualmente hay sensores mucho más potentes que detectan el desplazamiento de un objeto respecto a una posición relativa.

Temperatura y humedad

Con este otro sensor se obtienen datos de las condiciones de temperatura y humedad que se dan en el transcurso de una intervención. Según se recoge en la normativa del bloque quirúrgico: *"La temperatura del quirófano debe permanecer entre 22 y 26º C y la humedad relativa entre el 45 y 55 % según UNE 100713"*[21].

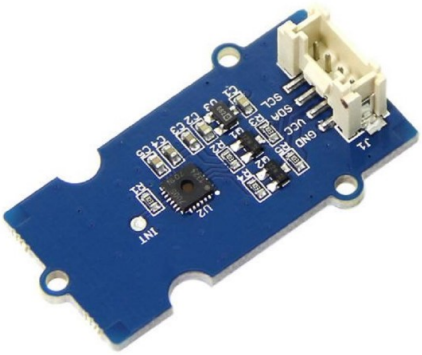


Figura 6: Sensor TH02

Concretamente, se ha utilizado el sensor de temperatura y humedad TH02 (ver figura 6) [22]. Los rangos de trabajo de este sensor son: para la temperatura entre 0 y 70 °C y para la humedad entre 0 y 80 %. Este sensor cuenta con la librería TH02_dev.h [23]

que facilita la programación de la lectura de los datos de temperatura y humedad. Esta librería, a su vez, requiere importar la librería Wire.h que permite la comunicación con dispositivos que se comunican mediante el bus I2C.

El bus I2C necesita únicamente dos cables: uno para el ciclo de reloj conectado al pin SCL (System Clock), y otro para el envío de los datos conectado al pin SDA (System Data). En el bus cada dispositivo dispone de una dirección única para poder acceder a cada dispositivo de los que tiene conectados de forma individual. Todos los sensores de temperatura TH02 escriben los datos en la misma dirección del bus, es por eso que desde una sola placa arduino sólo se pueden leer los datos de un sensor y ha sido necesario emplear una placa por cada sensor.

4.2. Base de datos MongoDB

MongoDB [24] es una base de datos no relacional de código abierto. Los datos se almacenan en documentos en formato BSON (Binary JavaScript Object Notation), similar a JSON (JavaScript Object Notation). Estos documentos a su vez se agrupan en colecciones. No es necesario conocer la relación entre los datos para almacenarlos y se pueden añadir con facilidad nuevos tipos de datos a los documentos de una colección así como más colecciones si fuera necesario ampliar el número de sensores o de información a almacenar.

Tiene integración con muchos lenguajes de programación entre los que se encuentran Python y C#, ambos empleados en este desarrollo.

Las características mencionadas proporcionan una escalabilidad y flexibilidad que la hacen adecuada para el sistema desarrollado.

4.3. Escenario 3D

4.3.1. Blender

Esta herramienta [25]se trata de un software para realizar modelado 3D libre, de código abierto y multiplataforma disponible para Linux, MacOS y Windows. Está escrito en C y C++, siendo posible crear extensiones con código Python. Con este software es posible diseñar gráficos tridimensionales de gran calidad. Gracias a su compatibilidad con Unity ha sido posible exportar el escenario al motor de videojuegos. La sala quirúrgica representada en este proyecto ha sido diseñada en su totalidad con Blender.

4.3.2. Unity

Unity [26] es un motor de videojuegos que permite desarrollar interfaces interactivas en 2D y 3D. Tiene soporte en Linux, MacOS y Windows, y emplea los lenguajes de programación C#, Unity y Boo. Para este proyecto ha sido empleado C# debido que este es el lenguaje más utilizado por la comunidad de usuarios de Unity y sus similitudes con Java facilitan su aprendizaje.

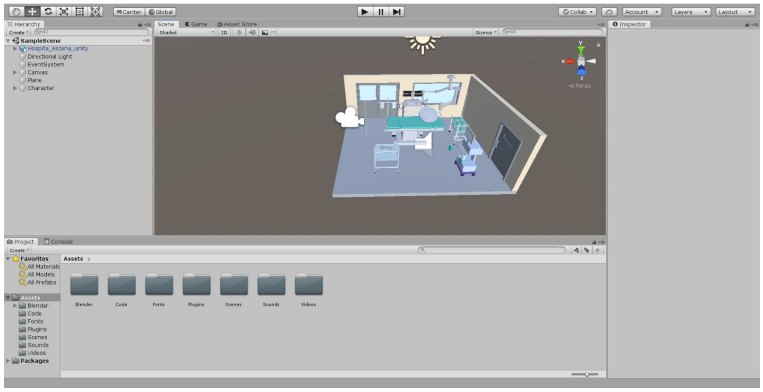


Figura 7: Unity

4.4. Lenguajes de programación

Python

Se trata de un lenguaje de programación multiparadigma y de código abierto que permite programación orientada objetos, funcional e imperativa [27]. Ha sido elegido porque actualmente es uno de los lenguajes de programación más empleados en el campo del *machine-learning*.

Librerías empleadas:

- **Pandas**: librería para la manipulación de datos de alto nivel [28].
- **Keras**: librería de código abierto para trabajar con redes neuronales y otros algoritmos de *machine-learning* [29].
- **scikit-learn**: Proporciona algoritmos de machine learning, herramientas de preprocesamiento de datos y métricas para evaluar los modelos entrenados [30].
- **PyMongo**: Esta librería contiene las funciones necesarias para trabajar con MongoDB desde Python [31].

C#

Este lenguaje de programación pertenece a la familia de los lenguajes de C, orientado a objetos [32]. Este es el lenguaje de programación que utiliza nativamente Unity.

Para poder conectar con la base de datos desde Unity, ha sido necesario instalar el driver de MongoDB para C#/.Net.

Arduino

Es un lenguaje de programación basado en C y C++, y proporciona la posibilidad de emplear comandos estándar de estos lenguajes en el código arduino [33]. Es bastante simple y se compone de, al menos, dos funciones necesarias setup(), que se encarga de la configuración, y loop(), la cuál contiene la parte del programa que se ejecutará cíclicamente. Este lenguaje se emplea en la programación de los microcontroladores Arduino y es compatible con otras plataformas hardware dada su naturaleza de software libre. Como editor ha sido empleado Arduino IDE (Integrated Development Environment).

5. Arquitectura del sistema

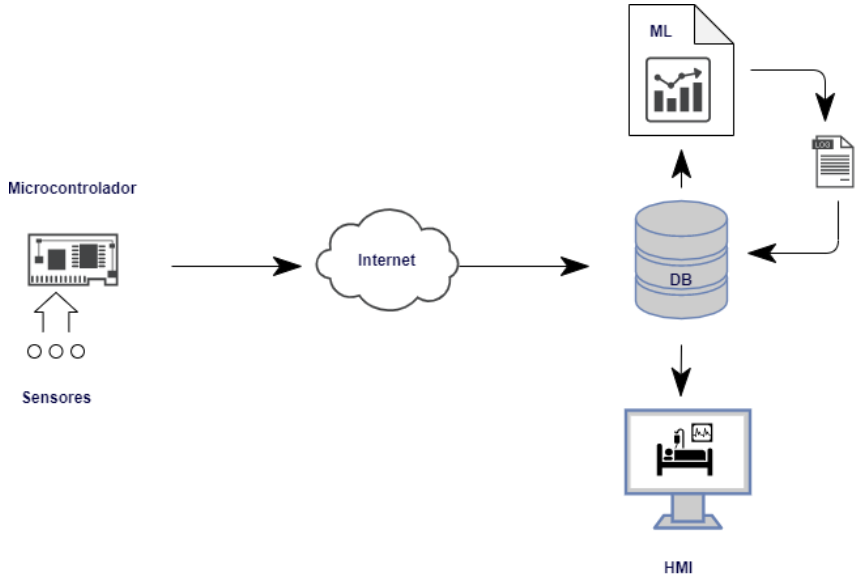


Figura 8: Arquitectura del sistema

La arquitectura del sistema y los módulos de los que consta para la recopilación, el almacenamiento e interpretación de los datos recogidos del escenario se detallan en este capítulo. Comenzando por el hardware, continuando por la gestión de los datos y la interfaz gráfica, y terminando con los algoritmos de machine learning.

5.1. Arduino y sensores

Las placas de Arduino son las unidades de procesamiento remotas que recogen la información de los sensores. Estos microcontroladores no cuentan con conexión inalámbrica, por lo que la información es enviada a la base de datos a través de un equipo al que se encuentra conectada.

El envío de datos de la placa Arduino al equipo al que se encuentra conectada (equipo 1) se realiza a través del puerto serie. Una vez el ordenador recibe los datos desde el microcontrolador, mediante un script de Python, el equipo 1 escucha el puerto serie

gracias a la librería serial, a su vez, este script mantiene una conexión abierta con Mongo para enviar la información a la base de datos. La base de datos se encuentra desplegada en otro equipo remoto (equipo 2) que hace las veces de servidor. Lo ideal sería que la información fuese transmitida a través de una LAN (Local Area Network) pero, debido a las restricciones físicas causadas por la COVID-19, en este proyecto se ha tenido que establecer una conexión a través de la WAN (Wide Area Network).

Los sensores de humedad y temperatura recogerán los valores del entorno. En cuanto a la posición de la camilla cobra especial importancia la situación de los sensores dentro de la habitación, los sónares se encontrarían en la mesa de operaciones tal y como se muestra en la figura 9.

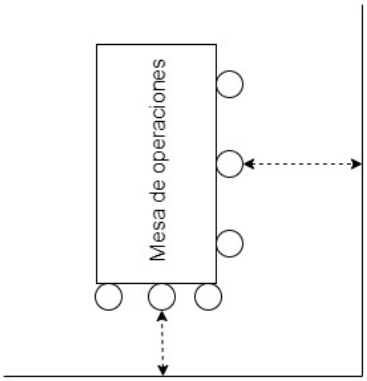


Figura 9: Disposición de los sónares en la camilla

5.2. Diseño de la base de datos

Dentro de MongoDB se ha creado una base de datos denominada “cps” y cuenta con cinco colecciones de datos:

- **packet_data**: Se almacenan los datos recogidos de las cabeceras de los paquetes tcp que contienen los datos de los sensores durante el desarrollo de la intervención.
- **packet_data_train**: Al igual que en packet data se guardan los datos de las cabeceras TCP durante la recogida de datos para entrenar el sistema.

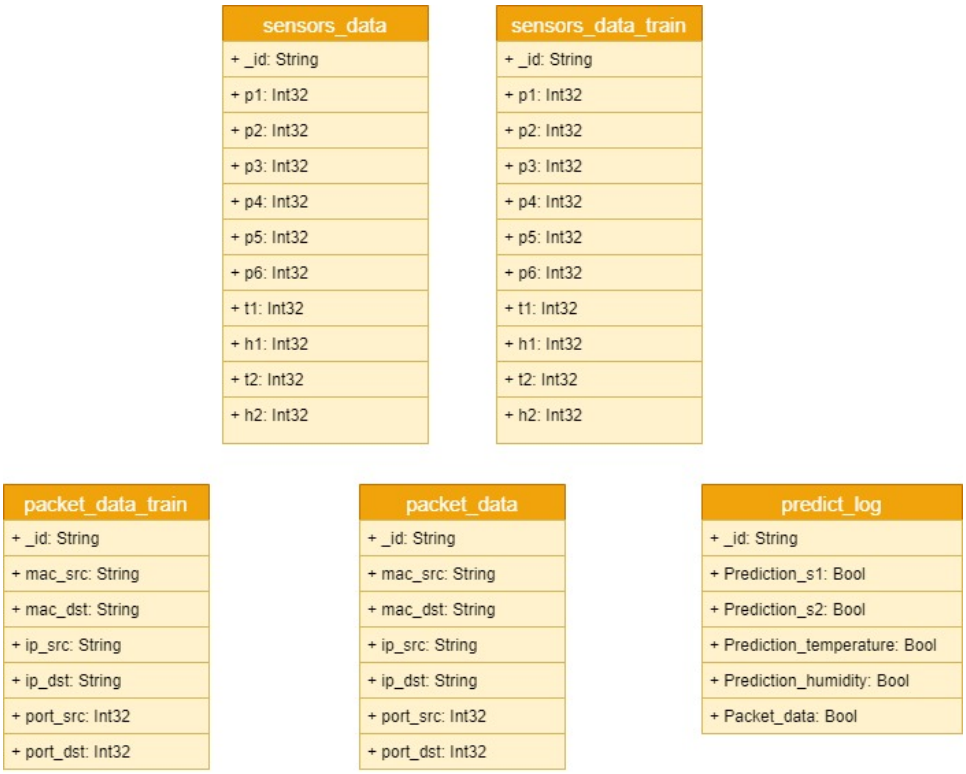


Figura 10: Colecciones de la base de datos

- **sensors_data**: Para guardar los datos recogidos por los sensores durante el desarrollo de la intervención.
- **sensors_data_train**: En esta colección se almacenan los datos de los sensores para el entrenamiento del algoritmo de machine learning.
- **predict_log**: Durante el desarrollo de la intervención los resultados de la clasificación de los datos que hacen los algortimos se almacenan en la base de datos.

En el sistema los procesos se comunican con la base de datos para actualizar la información de forma continua ya sea para actualizar la información de la base de datos o para obtener información actualizada. A continuación, se presentan los usuarios y los permisos con los que cuenta cada uno. Los permisos han sido asignados siguiendo el principio de mínimos privilegios.

- **admin**: Este será el usuario para el administrador de la base de datos que cuenta con todos los permisos.
- **arduino**: Desde el script de Python que envía las medidas recogidas por los sensores desde el equipo 1 a la base de datos por lo que requerirá acceder a la base de datos con permiso de escritura. Este permiso en MongoDB se denomina "readWrite".
- **machine-learning**: Por otro lado, en los scripts que ejecutan tanto el entrenamiento, como la clasificación del algoritmo establecen conexión con la base de datos. Para el entrenamiento será necesario el permiso de lectura, y para la clasificación tanto lectura como escritura para llevar el log de los resultados.
- **unity**: Para reflejar lo que está ocurriendo en el escenario real la interfaz gráfica actualizará la información del entorno consultando a la base de datos sobre las medidas de los sensores. También mostrará si se ha producido alguna anomalía consultando el log de las predicciones. Este usuario sólo necesitará el permiso de lectura, denominado "read" en MongoDB.
- **sniff**: Desde el equipo en el que se encuentra la base de datos se leerán todos los paquetes de datos TCP que insertan datos en cps para almacenar los datos de las cabeceras con los que se detectarán posibles ataques al sistema por parte de dispositivos desconocidos. Por lo tanto, también tendrá permiso "readWrite".

5.2.1. Comunicaciones con la base de datos

La base de datos es el eje central del intercambio de información entre los procesos que intervienen en el sistema como puede apreciarse en la figura 8.

En primer lugar, el equipo al que está conectado el microcontrolador envía los valores registrados por los sensores mediante un script de Python que son leídos del puerto serie del equipo. Por otro lado, el equipo en el que se encuentra la base de datos se ejecuta otro script de Python que obtiene la información de las cabeceras de los paquetes TCP que

van dirigidos a la base de datos, para evitar la lectura e inserción de datos por parte de dispositivos no autorizados.

El script de Python que ejecuta el algoritmo de detección de anomalías en las medidas tomadas y la comprobación de los datos de las cabeceras también se conecta a la base de datos para consultarlos. Posteriormente, una vez se ejecuta el algoritmo de detección, se guarda en la base de datos si ha detectado una anomalía o no.

La interfaz gráfica (Unity) consulta la información de la base de datos tanto de los sensores para actualizar la posición de la mesa de operaciones y los datos de temperatura y humedad que apareceran en las pantallas, como del log de anomalías para que quede registrado.

El hecho de que las comunicaciones se produzcan a través de la base de datos presenta ventajas e inconvenientes. Las ventajas son que toda la información queda registrada en la base de datos y que las partes podrían ejecutarse en distintos dispositivos atendiendo a los distintos recursos requeridos por cada proceso. Por ejemplo, la interfaz gráfica podría derivarse a cualquier dispositivo, tanto de fijo como móvil. Sin embargo, presenta desventajas como que hay que proteger correcta y eficientemente las comunicaciones, y que la información debe dar más saltos hasta llegar a su destino.

5.3. Diseño del escenario 3D

El diseño de la ventana de monitorización. En un sistema de supervisión el sentido de la vista es el sentido más explotado seguido del oído. En la elaboración del diseño de esta interfaz se ha tenido en cuenta, elaborando diseños de líneas simples y activando alarmas sonoras.

Blender

A la hora de realizar una representación de un proceso como es el caso de este TFG, lo recomendable es hacer un diseño minimalista y sin demasiado realismo en los objetos. Primero fueron modelados uno a uno los elementos de la sala de operaciones con Blender, basados en una imagen de un quirófano [34] y siguiendo las indicaciones del doctor. Los elementos presentes en el escenario y visibles en la figura 11 son:

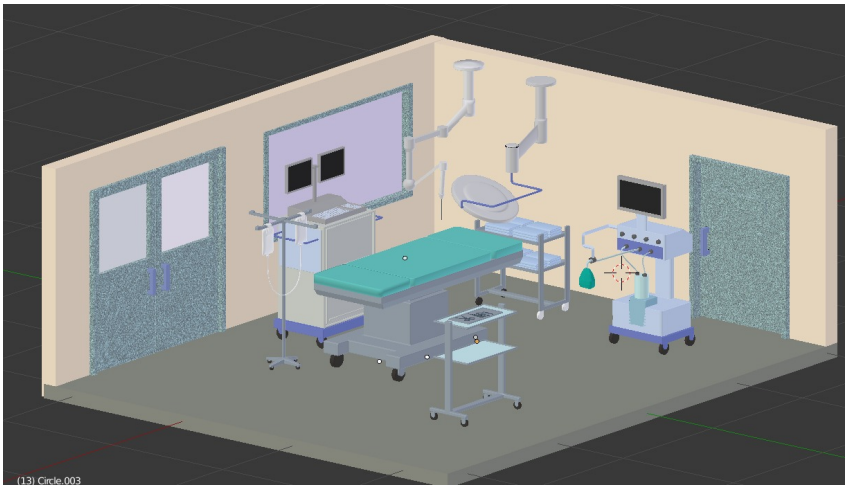


Figura 11: Sala de operaciones

- **Puerta principal**: esta sería la puerta por la que se introduce al paciente.
- **Puerta de sucio**: e través de la cual extraen todo el material y residuos al finalizar la operación.
- **Mesa de operaciones**: camilla en la que se sitúa al paciente durante la intervención.
- **Torre de gases y electrocardiograma**: maquinaria para la ventilación mecánica del paciente y administración de anestesia. Y electrocardiograma para visualizar en todo momento las constantes vitales del paciente.
- **Carrito instrumentación**: en este carrito se deposita el instrumental médico.

- **Carrito auxiliar con gasas**: carrito con gasas preparadas para la intervención.
- **Lámpara**: una lámpara para tener una correcta iluminación de la zona sobre la que se va a realizar la intervención.
- **Portasueros**: percha para depositar las botellas con suero y medicación.
- **Un par de monitores y un teclado**: en este par de monitores se reflejarán los valores de humedad y temperatura durante la intervención.
- **Robot de operaciones**: se ha dibujado una especie de robot quirúrgico.

Posteriormente, este escenario fue importado a Unity y fueron añadidas las texturas y comportamientos para proporcionar la actividad al escenario.

Unity

Al entrar en la aplicación de Unity se presenta un menú con dos opciones para entrar en la representación gráfica del escenario, una para poder visualizar el escenario mediante el movimiento de la cámara y otra que lleva a la visualización del estado del quirófano en tiempo real.

Al modelar el comportamiento del escenario, en primer lugar, fueron añadidas las texturas que dan color a la interfaz. Cabe destacar, que son texturas simples con el objetivo de evitar la distracción de los usuarios, además de ser menos costosas en ejecución. Por otro lado, para que los objetos no puedan ser atravesados en las colisiones se añadió el objeto Box Collider a las paredes, el suelo, la camilla y demás mobiliario que pueda colisionar con otros objetos. Por otro lado, para que la camilla tenga los comportamientos de un objeto regido por las leyes de la física, pudiendo recibir fuerzas y moverse de una manera realista, le fue asignado el componente Rigidbody.

Para la visualización de las anomalías en el proceso de monitorización se han implementado mecanismos para la gestión de anomalías por vigilancia y por excepción. La

gestión por vigilancia se realiza gracias al movimiento de la camilla de acuerdo con los valores recogidos por los sensores de ultrasonido y la visualización de las medidas de temperatura y humedad que aparecen reflejadas en las pantallas de la escena. En cuanto a la gestión por excepción, se ha establecido una alarma visual, un texto en color rojo indicando de dónde procede la anomalía, y una alarma sonora. El color rojo indica peligro en consonancia con los estereotipos y empleado por convención, aplicando el principio de coherencia [35].

En Unity para establecer el comportamiento de un objeto se le asigna un script en C#. Todos los scripts C# empleados derivan de la clase Monobehaviour y ejecutan las funciones Start() y Update(). La función Start() se ejecuta una sola vez al comenzar la escena, es en esta función dónde se establece la conexión con la base de datos. Update() se ejecuta en cada frame de la representación consultando los últimos datos introducidos en la base de datos con el objetivo de actualizar la información representada en la escena.

5.4. Machine Learning

El *machine-learning* [36] es un ámbito de la Inteligencia Artificial que consiste en encontrar patrones a partir de los datos. En este TFG gracias a los algoritmos no supervisados se han elaborado modelos que representan las medidas de los sensores en condiciones normales del sistema, ya que los ataques a sensores ponen en cuestión las mediciones del escenario y con ello la integridad física del personal sanitario y los pacientes.

Cuando se inicia la monitorización el algoritmo va comprobando todos los datos de entrada cada cierto período de tiempo, si detecta alguna anomalía queda reflejado en la base de datos en una colección de logs.

En el siguiente capítulo se exponen con detalle los algoritmos para la detección de intrusiones y de anomalías, así como el proceso de preprocesamiento de los datos con los que han sido entrenados los algoritmos.

6. Detección de anomalías e intrusiones

Para la detección de valores anómalos y el acceso de intrusos a la base de datos han sido empleados dos mecanismos distintos que se describen en este capítulo. También se detallan los datos utilizados y el preprocesamiento aplicado para la elaboración de los modelos.

Trasladando ambos problemas de la vida real al *machine-learning*, se observa que se trata de problemas de clasificación binaria entre valores normales y valores anómalos. Puesto que los ataques más repetidos y difíciles de detectar son aquellos producidos por *insiders* o intrusos. Por otro lado, el ataque a sensores y actuadores puede producir daños físicos en los escenarios en los que se encuentran. Por estos motivos, la detección de anomalías en los valores de los sensores es de vital importancia.

6.1. Detección de intrusiones

Gracias a la librería scapy [37], herramienta para la manipulación de paquetes de red, capturando el tráfico de red se pueden identificar las variables para identificar algún paquete sospechoso. En este caso, la detección de intrusiones está enfocado a los ataques a la base de datos, por tanto se han analizado los paquetes TCP dirigidos a la base de datos Mongo.

Para la detección de intrusiones se ha aplicado una *whitelist* (listas blancas) ya que es el sistema que mejores resultados obtiene. Por ejemplo, cuando una red neuronal predice un campo de la cabecera, cuyos valores suelen tener poca variabilidad, la efectividad de la red es menor que la de la *whitelist* [38]. Las *whitelists* han sido empleadas con éxito en sistemas de control industrial encargados de la monitorización y control de dispositivos interconectados generalmente conectados en entornos conocidos [39].

Una *whitelist* consiste en un listado con las direcciones IP, direcciones MAC, o direccio-

nes de correo electrónico permitidos; cualquiera que no esté en dicho listado es bloqueado. Los tipos de datos que almacenan estas listas son personalizables según las necesidades de la aplicación.

De los datos disponibles en la capa TCP en esta *whitelist* se han empleado los siguientes:

- **IP origen**: Identifica al equipo del cual proceden los datos.
- **IP destino**: Identifica al equipo al que se dirigen los datos.
- **MAC origen**: Es la dirección física del equipo del que proceden los datos.
- **MAC destino**: Es la dirección física del equipo al que se dirigen los datos.
- **Puerto de destino**: El puerto de origen no es fijo y va variando, por lo que no ha sido considerado.

Se elabora una lista con las IPs, direcciones MAC y puerto de destino que tienen acceso legítimo a la base de datos. En caso de detectar algún valor que no esté permitido se lanza una alerta. Es importante tener en cuenta que para que este sistema funcione correctamente con las direcciones IP, éstas deben ser estáticas. Con las direcciones MAC no hay problema pues todas son estáticas. El puerto de destino de la base de datos también es fijo.

6.2. Detección de anomalías

El algoritmo debe ser capaz de distinguir si los datos de entrada están dentro de lo esperado o representan una anomalía. El algoritmo implementado para ello es de aprendizaje no supervisado puesto que los datos recogidos en el entorno de aplicación no están etiquetados. Lo que sí se conoce es el conjunto de datos de entrenamiento en condiciones normales, es decir, no presentan anomalías. La tarea se ha centrado en detectar la

estructura de dichos datos para poder distinguir entre entradas normales y anómalas. La implementación elegida no sólo nos permite detectar que está ocurriendo una anomalía en el sistema, sino que también diagnostica de dónde proviene.

Las fases de desarrollo de un algoritmo de *machine-learning* son: recopilación de datos, preparación de los datos, entrenamiento del modelo y validación del mismo.

6.3. Algoritmo

Algoritmo planteado está basado en [38] y se trata de un tipo de arquitectura de redes neuronales denominada autoencoder.

Las redes neuronales se basan en las neuronas humanas, una neurona artificial realiza cálculos y se conecta con otras neuronas artificiales para realizar alguna tarea específica. Las distintas combinaciones de neuronas artificiales se denominan arquitecturas. Dentro de las arquitecturas tienen tres partes diferenciadas siempre presentes en una arquitectura: capa de entrada (*input*), capa de salida (*output*) y capas ocultas intermedias (*hidden*). Cuantas más capas tenga un modelo, mayor es el coste computacional.

El autoencoder mapea la entrada codificándola internamente para extraer características en datos de alta dimensionalidad, y a partir de la codificación trata de reconstruir la observación de entrada. En el modelo empleado, los datos de entrada (u observaciones) son los datos recogidos en una ventana de tiempo, como serían los datos en el recuadro de la figura 12.

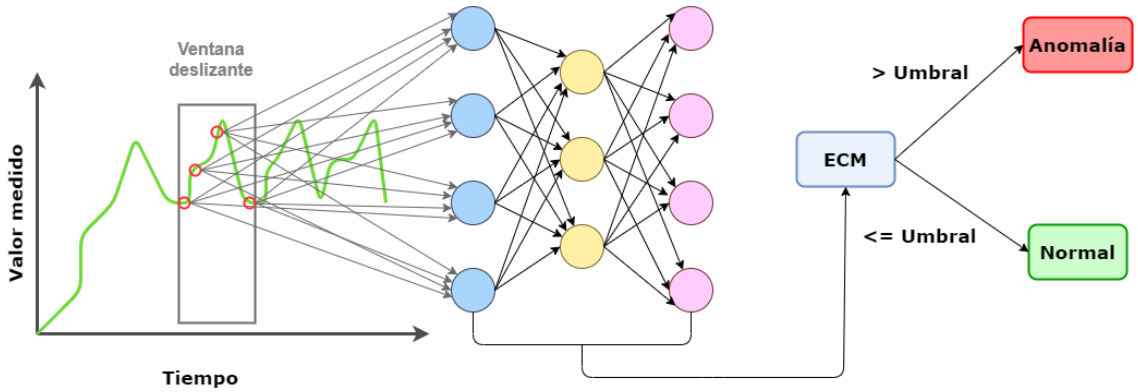


Figura 12: Diagrama del algoritmo

Para las observaciones normales la reconstrucción de los datos se llevará a cabo con precisión, mientras que para las observaciones anómalas se detectará un error entre la entrada y la salida por encima de un umbral establecido empíricamente. Para medir la diferencia entre la entrada y la salida se ha empleado el ECM (Error Cuadrático Medio).

Se ha entrenado un modelo diferente para cada tipo de medida, es decir, en este caso se han empleado cuatro modelos diferentes, uno para la temperatura, uno para la humedad y dos para la posición de la camilla, uno para cada grupo de sensores de los laterales perpendiculares. Esto presenta la ventaja de que es posible detectar de qué medida o medidas procede la anomalía.

Para el entrenamiento y validación de los modelos se ha empleado la librería TensorFlow en un equipo con procesador intel core i5-7200U, 8 GB RAM.

Otro motivo para la elección de este algoritmo ha sido el requisito de la detección en tiempo real, ya que los tiempos de respuesta de este tipo de algoritmos son muy bajos.

6.3.1. Preprocesamiento de los datos

Una vez recogidos los datos de los sensores y almacenados en la base de datos, han sido descargados para preparar la entrada de los modelos y realizar el entrenamiento. Se ha

tratado de recoger un conjunto de datos en condiciones normales, es decir, un dataset sin anomalías con el que entrenar el modelo para obtener un error de reconstrucción mínimo. Debido a los fallos del sensor puede haber alguna anomalía, por lo que se eliminan valores erróneos que están fuera del rango de medidas del sensor.

Normalización de los datos

En primer lugar, se normalizan los datos de entrada reduciéndolos a un rango de 0 a 1 con el objetivo de concentrar las entradas en dicho rango permitiendo un tratamiento de los datos más sencillo.

De las técnicas de normalización se ha seleccionado MinMax, que sigue la siguiente expresión:

$$valor_{normalizado} = \frac{valor - x_{min}}{x_{max} - x_{min}} \tag{2}$$

Siendo x_{min} el valor mínimo que puede tomar un atributo y x_{max} el máximo.

Los valores mínimos y máximos son los mínimos y máximos medidos por los sensores.

- **Temperatura**: mínimo 0 y máximo 70 °C.
- **Humedad**: mínimo 0 y máximo 80 %.
- **Sónares**: mínimo 0 y máximo 200 cm.

Este método de normalización ha sido escogido puesto que otros métodos que utilizan la desviación típica y la media son muy sensibles a datos anómalos pudiendo distorsionar los datos [40]. En este caso, la normalización sobre los datos de entrenamiento quedaría como se muestra en las figuras 13 y 14 y 15, los datos se reescalan sin pérdida de información.

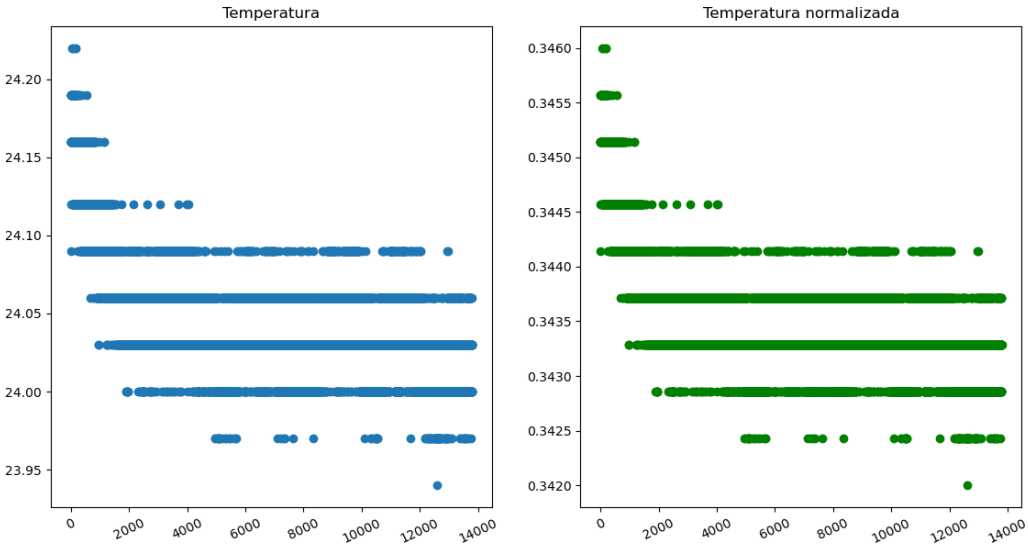


Figura 13: Normalización de la temperatura

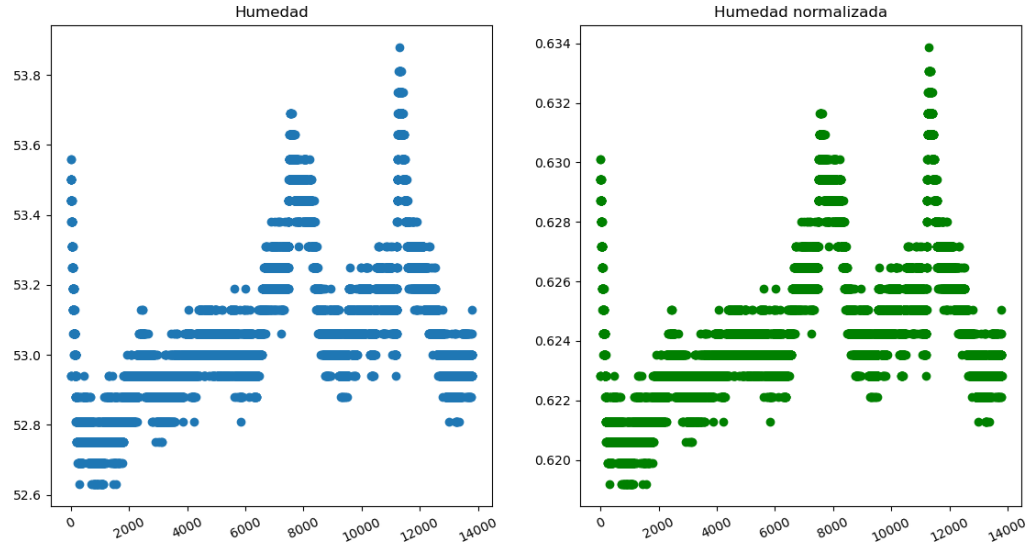


Figura 14: Normalización de la humedad

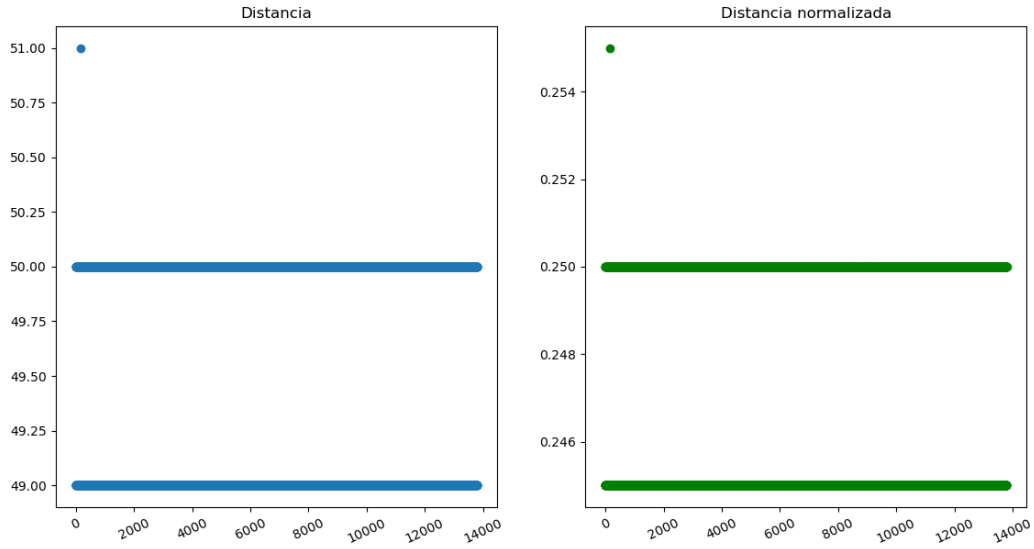


Figura 15: Normalización Sonar

Creación de secuencias de datos

Una vez se han obtenido los valores normalizados, se procede a la creación de secuencias de datos por intervalos de tiempo como entrada al autoencoder. Las secuencias se crean con una ventana deslizante de los valores recogidos de cada sensor durante un período de tiempo de 10 segundos aproximadamente. Tenemos unas 7 entradas de datos por segundo en la base de datos, por lo que cada entrada a cada modelo será de 70 medidas del sensor correspondiente.

6.3.2. Construcción del modelo y validación

Entrenamiento

Se han generado 4 modelos: un modelo para los datos de temperatura, otro para la humedad, uno para el grupo de sónares de un lateral y otro para el grupo de sónares que

están en el lateral perpendicular.

Todos los modelos presentan una estructura proporcional a los datos de entrada con dos capas intermedias siguiendo un esquema como el que presenta la figura 16.

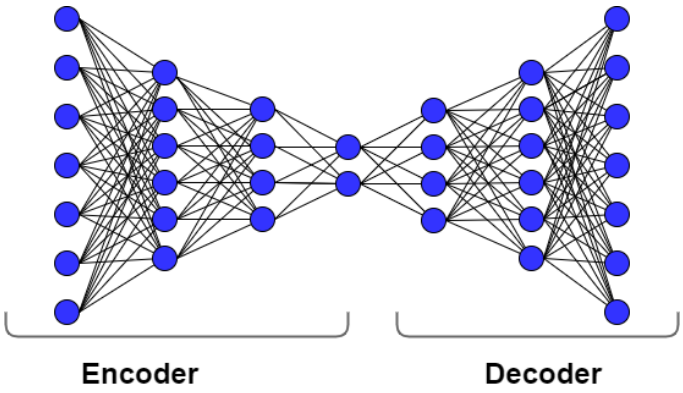


Figura 16: Estructura Autoencoder

- *Encoder*: codifica la entrada a una dimensionalidad menor, generando nuevas características. La capa intermedia puede considerarse una representación de la entrada con una menor dimensionalidad.
- *Decoder*: utilizando la capa intermedia trata de reconstruir la entrada a partir de la codificación generada.

Como funciones de activación se han empleado ReLU y la función sigmoide, dado que sus valores de salida se encuentran en el intervalo [0, 1], al cual han sido normalizados los datos. Para la función de reconstrucción o pérdida se ha empleado el Error Cuadrático Medio (ECM).

Los modelos han sido entrenados empleando el 70 % de los datos para entrenar y el 30 % para validación.

Establecimiento del umbral

Para detectar las anomalías, tras entrenar la red, el siguiente paso es el establecimiento de un umbral de error producido entre la entrada y la salida del algoritmo para detectar si la entrada se trata o no de una observación anómala.

El umbral se ha establecido según el modelo, con el error máximo o el error que se encontraba entre los percentiles 90 y 99. Esto dependerá de la calidad del sensor. Para el cálculo de este error también ha sido empleado el Error Cuadrático Medio.

$$ECM = \frac{1}{n}\sum_{i=1}^{n} X_i - \bar{X} \tag{3}$$

6.3.3. Resultados del algoritmo

Al tratarse de un algoritmo no supervisado no es posible emplear métricas como F1 Score, Precision o Recall, en lugar de eso se han graficado los resultados obtenidos señalando en rojo las anomalías detectadas.

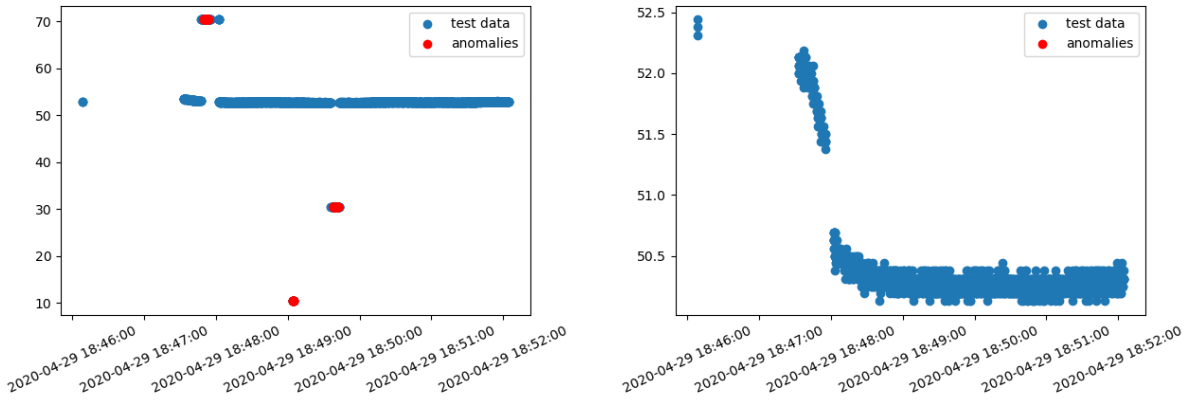
Para que un valor sea considerado anómalo se ha empleado un sistema por votación. Puesto que los sensores pueden ser atacados o dejar de funcionar debido a algún tipo de fallo se disponen de, al menos, dos sensores de cada tipo, se considerará que una observación es anómala si las observaciones de entrada del 50 % de los sensores más uno, son clasificadas como anómalas. En la representación gráfica se considerará que un valor concreto es anómalo si en todas las observaciones en las que se encuentra se obtiene una salida anómala. En otras palabras, durante todo el tamaño de ventana debe producirse una salida anómala.

En todo momento se ha tenido muy presente que el algoritmo debe proporcionar fiabilidad, escalabilidad y modularidad. Por ello, se han implementado modelos independientes para cada tipo de sensor siendo posible aumentar o disminuir la cantidad y tipos

de sensores.

A continuación, se muestran resultados de aplicar los modelos entrenados a datos con y sin anomalías.

Comenzando con los datos de humedad, la figura 17 podemos observar que en 17a hay algunas anomalías en valores que se desvían considerablemente, mientras que en 17b no hay anomalías puesto que los valores se encuentran estables entre 50 y 52.5.

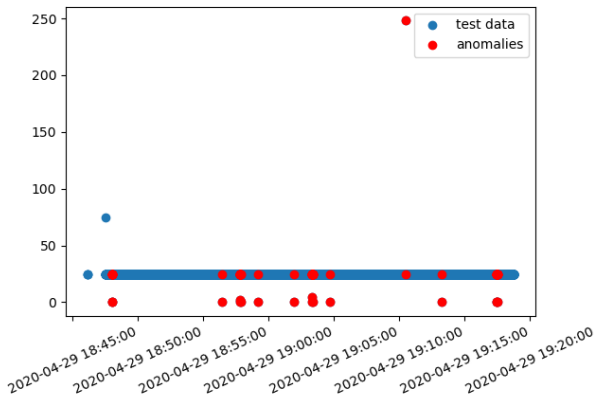


(a) Valores de humedad con anomalías

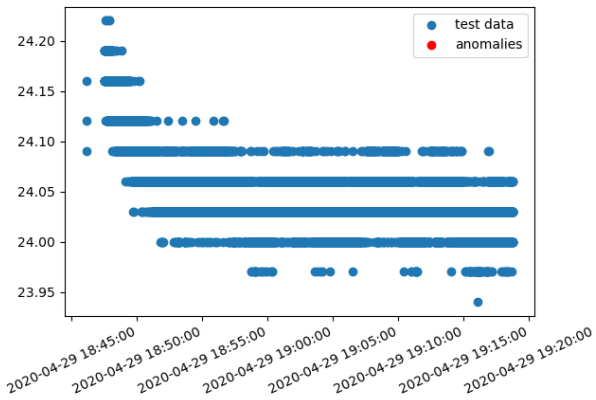
(b) Valores de humedad sin anomalías

Figura 17: Anomalías en datos de humedad

Continuando con las temperaturas, en la figura 18 tenemos que los valores sin anomalías (17b) se concentrados en el intervalo 23.9-24.25. Mientras que en los valores con anomalías (17a) se detectan valores de 0°C y uno de 250°C. Estos valores ocasionan grandes desviaciones en los datos, por lo que son considerados anomalías.



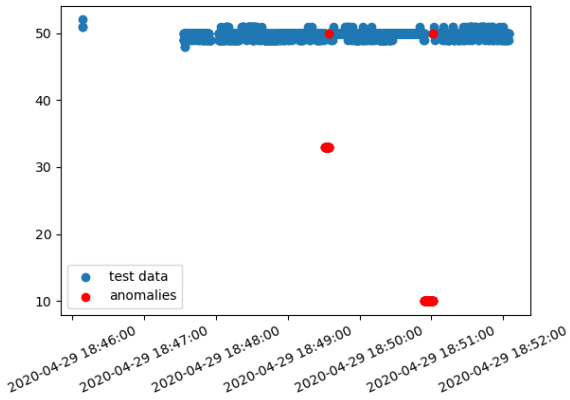
(a) Valores de temperatura con anomalías



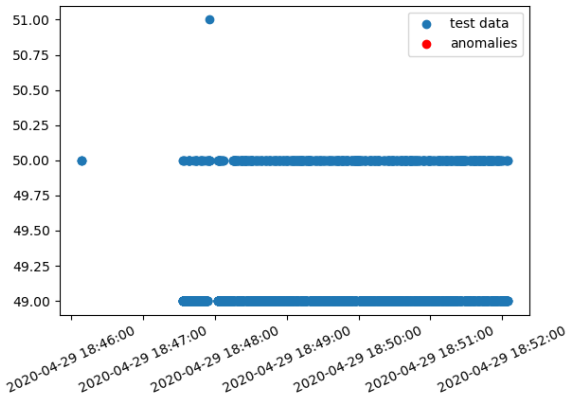
(b) Valores de temperatura sin anomalías

Figura 18: Anomalías en datos de temperatura

Por último, en la figura 19 se presentan los resultados obtenidos de testar el modelo generado para uno de los grupos de sónares. En los valores con anomalías (19a) las desviaciones son considerables mientras que en los valores sin anomalías (19b) hay un valor aislado que presenta una pequeña desviación que no es considerado anomalía puesto que puede deberse a un simple fallo de precisión del sensor.



(a) Valores de distancia con anomalías



(b) Valores de distancia sin anomalías

Figura 19: Anomalías en datos del grupo de sónares

7. Conclusiones y trabajo futuro

En este último apartado se presenta una discusión sobre los resultado obtenidos, así como el trabajo futuro.

7.1. Problemas encontrados y soluciones

La primera dificultad encontrada fué el número de sensores que había que conectar a la placa Arduino UNO. En dicha placa hay 14 pines de E/S digitales, para cada sónar son necesarios 2 pines digitales y para cada sensor de humedad y temperatura se necesitan dos pines SCL (Serial Clock) y SDA (Serial Data), pues siguen el protocolo I2C.

El protocolo I2C es un protocolo síncrono con el que un dispositivo maestro se comunica con uno o varios dispositivos esclavos para el envío de datos empleando el direccionamiento en memoria, a diferencia de otros protocolos que realizan selección del esclavo [41]. Con el protocolo I2C los dispositivos se comunican utilizando estas dos entradas, se pueden conectar varios dispositivos siempre y cuando cada uno escriba en una dirección de memoria distinta. Aquí es dónde encontramos la principal dificultad, pues al tener tres sensores iguales no se podían usar en la misma placa puesto que empleaban la misma dirección de memoria, había que utilizar una placa por cada sensor para tenerlos funcionando a la vez. Como se contaba con la posibilidad de utilizar una segunda placa, se emplearon dos de los tres sensores de humedad y temperatura.

Por otro lado, a la hora de escoger los sensores que se emplearían en este TFG, se trató de emplear sensores para medir el pulso y la presión ambiental. Para el sensor de pulso se compraron sensores que resultaron no funcionar. En cuanto al sensor para la presión ambiental se intentó adquirir un sensor que además de medir la temperatura y la humedad, también medía la presión, pero por motivos de stock no fué posible.

La precisión y la calidad de los datos medidos es algo crucial para realizar un buen

análisis, en este sentido, debido a que se trata de un TFG la calidad de los sensores es más baja y se ha tratado de obtener datos con la mayor calidad posible. En el caso de los sónares debían estar muy bien colocados para evitar interferencias de ultrasonidos de unos sensores a otros.

En la industria, los sistemas ciberfísicos se encuentran aislados en una LAN y protegidos por un firewall, en este proyecto dado que es un proyecto en grupo, las comunicaciones de los microcontroladores con la base de datos han tenido que darse a través de una WLAN.

También hay distintos tipos de normalizaciones, de las cuáles la que mejor codificaba los datos era MinMax. Para que al normalizar los valores de los sensores se encontrasen entre 0 y 1 ha habido que eliminar outliers que excedían el rango de valores medidos por el sensor debido a errores de medida.

Por último, destacar la dificultad añadida del desarrollo de este TFG grupal dada la situación excepcional generada por la COVID-19. Para el desarrollo común se han empleado todos los medios técnicos al alcance como videoconferencias, correo electrónico y sistemas de gestión de archivos compartidos.

7.2. Conclusiones

El internet de las cosas es una realidad, por ello la detección de anomalías y la protección de los sistemas ciberfísicos es crucial en cualquier aplicación. La seguridad de estos sistemas cobra especial importancia cuando se trata de infraestructuras críticas, como el caso que se aborda en este trabajo, para asegurar que los procedimientos implicados sean efectivos y seguros. Este proyecto ha sido una primera implementación de un sistema cuyos objetivos eran:

- Tener una simulación de una sala de operaciones en tiempo real mediante una interfaz gráfica en 3D.

- Detectar anomalías en las medidas proporcionadas por sensores.
- Detectar intrusiones en el sistema.

Durante todo el análisis y desarrollo del sistema se ha tenido presente que éste debía ser fiable y escalable. Este sistema de monitorización puede ampliarse añadiendo nuevos parámetros de los que detectar anomalías como por ejemplo las constantes vitales del paciente que está siendo operado. Podría emplearse tanto para monitorizar intervenciones reales como para simularlas y servir de programa de entrenamiento para estudiantes de las ramas sanitarias a través de una interfaz amigable.

Para tener una buena detección de anomalías es crucial obtener un dataset con los valores que se consideren correctos para una situación concreta en una sala de operaciones. En esto influyen tanto la calidad de las medidas de los sensores, como el conocimiento del sistema. En cuanto a la detección de intrusiones, es un campo en desarrollo y para alcanzar un sistema más completo y efectivo aún queda mucho trabajo de investigación para llegar a cubrir más tipos de ataques. En este caso se ha contado con la oportunidad de conocer en primera persona cómo son las instalaciones en un hospital gracias a la colaboración del cliente.

A nivel académico, este proyecto ha supuesto un esfuerzo por conjugar muchos conocimientos adquiridos a lo largo del grado y el aprendizaje de nuevas tecnologías tan variadas como las que se presentan en este sistema, así como el desarrollo de la capacidad de adaptación y ser flexible a la hora de afrontar problemas y encontrar soluciones. Otro reto importante ha sido la coordinación y el trabajo en equipo.

7.3. Trabajo futuro

Las posibles mejoras y ampliaciones del sistema implementado pueden ser las siguientes:

- Sustituir los microcontroladores sin conexión inalámbrica por otros que sí la tengan como por ejemplo las placas Arduino maker [42] o el microcontrolador ESP8266 [43].
- Extender la monitorización y detección de anomalías con más sensores como la presión, factor ambiental importante en una sala de operaciones tal y como nos comentaba el doctor en la entrevista.
- Añadir la simulación de una operación con personajes en la interfaz. Además, se podrían añadir gráficas en tiempo real que permitan visualizar el valor de los sensores.
- La detección de intrusiones, pese a que el mecanismo implementado es efectivo, en esta aplicación podría mejorarse y complementarse con la aplicación de algoritmos y un preprocesamiento más elaborado de los datos que permitan detectar más tipos de ataques como podrían ser los retrasos en la entrega de paquetes.
- La versión Community de MongoDB no permite tener un archivo de logs, lo cual sería de gran interés en una infraestructura crítica como esta. Para ello, habría que emplear una versión más avanzada como sería Enterprise.

Referencias

[1] National Sciety Foundation (NSF). Cyber-physical systems (CPS). Accedido el 20-07-2020, https://www.nsf.gov/pubs/2014/nsf14542/nsf14542.htm.

[2] C. Ten, G. Manimaran, and C. Liu. Cybersecurity for critical infrastructures: Attack and defense modeling. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(4):853–865, 2010.

[3] Y. Xu, D. Tran, Y. Tian, and H. Alemzadeh. Poster abstract: Analysis of cybersecurity vulnerabilities of interconnected medical devices. pages 23–24, 09 2019.

[4] G. McGraw. Software security. *IEEE Security Privacy*, 2(2):80–83, 2004.

[5] Wang Lidong and Wang Guanghui. Big data in cyber-physical systems, digital manufacturing and industry 4.0. *International Journal of Engineering and Manufacturing*, 6:1–8, 07 2016.

[6] J. C. Llinás, J. Meléndez, and J. Ayza. Sistemas de supervisión: introducción a la monitorización y supervisión experta de procesos : métodos y herramientas. 2000.

[7] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic. Cyber-physical systems: The next computing revolution. In *Design Automation Conference*, pages 731–736, 2010.

[8] J. Flores. Practicada la primera operación teleasistida con 5G. Accedido el 25-07-2020, https://www.nationalgeographic.com.es/ciencia/actualidad/practicada-primera-operacion-teleasistida-5g_13948, Febrero 2020.

[9] R. Eveleth. Los cirujanos que operan a cientos de kilómetros de distancia. Accedido el 20-06-2020, https://www.bbc.com/mundo/noticias/2014/05/140520_vert_fut_salud_cirujano_a_distancia_gtg, Mayo 2014.

[10] C. Alcaraz, L. Cazorla, and G. Fernandez. Context-awareness using anomaly-based detectors for smart grid domains. In *9th International Conference on Risks and*

Security of Internet and Systems, volume 8924, pages 17–34, Trento, 04/2015 2015. Springer International Publishing, Springer International Publishing.

[11] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, 2016.

[12] F. Sabahi and A. Movaghar. Intrusion detection: A survey. In *2008 Third International Conference on Systems and Networks Communications*, pages 23–26, 2008.

[13] E. Hossain, P. L. Bannerman, and D. R. Jeffery. Scrum practices in global software development: A research framework. In Danilo Caivano, Markku Oivo, Maria Teresa Baldassarre, and Giuseppe Visaggio, editors, *Product-Focused Software Process Improvement*, pages 88–102, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[14] Altassian. Trello. Accedido el 22-04-2020 `https://trello.com/es`.

[15] D. J. Powell. Kanban for lean production in high mix, low volume environments. *IFAC-PapersOnLine*, 51(11):140 – 143, 2018. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[16] Instituto Nacional de Estándares y Tecnología (NIST). Marco para la mejora de la seguridad cibernética en infraestructuras críticas. Accedido el 10-05-2020 `https://www.nist.gov/system/files/documents/2018/12/10/frameworkesmellrev_20181102mn_clean.pdf`.

[17] US. Department of Health & Human Services. Reglas HIPAA. Accedido el 10-05-2020 `https://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/administrative/combined/hipaa-simplification-201303.pdf`.

[18] US. Department of Health & Human Services. HIPAA for professionals. Accedido el 30-07-2020, `https://www.hhs.gov/hipaa/for-professionals/index.html`.

[19] Arduino. Accedido el 21-05-2020 `https://www.arduino.cc/`.

[20] E. Freaks. Datasheet hc- sr04. Accedido el 16-05-2020, https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf.

[21] Ministerio de Sanidad y Política Social. Bloque quirúrgico: Estándares y recomendaciones. Accedido el 20-05-2020, https://www.mscbs.gob.es/organizacion/sns/planCalidadSNS/docs/BQ.pdf.

[22] Seed Studio. Temperature & humidity sensor TH02. Accedido el 20-05-2020, https://files.seeedstudio.com/wiki/Grove-TemptureAndHumidity_Sensor-High-Accuracy_AndMini-v1.0/res/TH02_SENSOR.pdf.

[23] Seed Studio. Grove temperature & humidity th02 library. Accedido el 20-05-2020, https://github.com/Seeed-Studio/Grove_Temper_Humidity_TH02.

[24] Mongodb. Accedido el 21-05-2020 https://www.mongodb.com/.

[25] T. Roosendaall. Blender. Accedido el 21-05-2020 https://docs.blender.org/manual/es/dev/getting_started/about/introduction.html.

[26] Unity. Accedido el 21-05-2020 https://unity.com/es.

[27] Python. Accedido el 21-05-2020 https://www.python.org/.

[28] Pandas Development Team. Librería Pandas. Accedido el 20-05-2020, https://pandas.pydata.org/pandas-docs/stable/index.html.

[29] F. Chollet. Librería Keras. Accedido el 23-05-2020, https://keras.io/.

[30] D. Cournapeau. Librería Scikit-Learn. Accedido el 23-05-2020, https://scikit-learn.org/stable/.

[31] MongoDB. Librería pymongo. Accedido el 23-05-2020, https://pymongo.readthedocs.io/en/stable/.

[32] Especificación del lenguaje c #. Accedido el 21-05-2020 https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/language-specification/introduction.

[33] Lenguaje arduino. Accedido el 21-05-2020 https://www.arduino.cc/reference/en/.

[34] Macrovector. Operation room equipment isometric composition vector image. Accedido el 05/03/2020, https://www.vectorstock.com/royalty-free-vector/operation-room-equipment-isometric-composition-vector-19200524.

[35] A. Rodríguez Penin. *Sistemas SCADA*. Marcombo, Barcelona, 3 edition, 2012.

[36] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung. A survey on security threats and defensive techniques of machine learning: A data driven view. *IEEE Access*, 6:12103–12117, 2018.

[37] P. Biondi, G.Valadon, P. Lalet, and G. Potter. Scapy. Accedido el 20-06-2020, https://scapy.net/.

[38] S. Kim, W. Jo, and T. Shon. Apad: Autoencoder-based payload anomaly detection for industrial ioe. *Applied Soft Computing*, 88:106017, 2020.

[39] I. Garitano, M.Iturbe Urretxa, U. Zurutuza Ortega, and R. Uribeetxeberria. Sistema visual de monitorización de seguridad de flujos de red industriales. 2015.

[40] S. Morante Cendrero. Precauciones a la hora de normalizar datos en data science. Accedido el 29-06-2020, https://empresas.blogthinkbig.com/precauciones-la-hora-de-normalizar/.

[41] E. Crespo. Aprendiendo Arduino. Protocolo I2C. Accedido el 27-07-2020, https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/.

[42] Arduino.cc. Arduino maker. Accedido el 28-07-2020, https://store.arduino.cc/arduino-mkr-wifi-1010.

[43] J. López. Arduino maker. Accedido el 28-07-2020, https://hardzone.es/reportajes/tema/esp8266-2n2222-arduino/.

Anexos

A. Manual de Usuario

Al abrir el ejecutable de unity donde se encuentra la interfaz de la aplicación se muestran, como aparece en la figura 20, dos opciones: explorar el escenario e iniciar detección de anomalías. También aparece un botón para salir del programa.

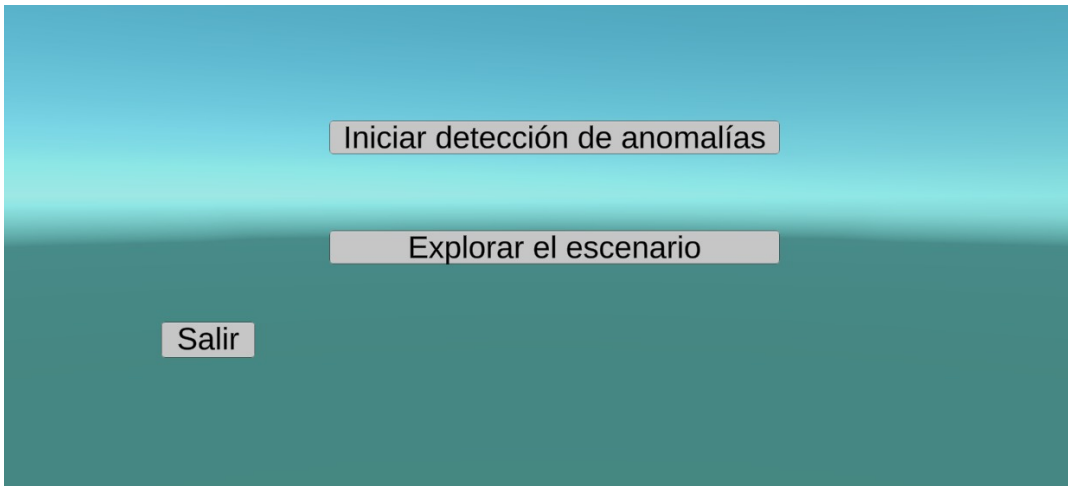


Figura 20: Página de inicio en Unity

Explorar el escenario

El motivo de la inclusión de esta opción en la aplicación es la posibilidad de poder explorar el escenario creado sin necesidad de tener la base de datos funcionando.

Al clickar en esta opción de explorar el escenario, nos llevará a la representación gráfica de la sala quirúrgica en la que se podrá navegar moviendo la cámara de la siguiente manera:

- Con el ratón se podrá mover la cámara en la dirección que indiquemos con el cursor.

- Con las teclas **W** para avanzar, **S** moverse hacia atrás, **D** hacia la derecha y con **A** a la izquierda.
- Al pulsar **esc** se volverá al menú principal.

Iniciar detección de anomalías

En la opción iniciar detección de anomalías se iniciará una conexión con la base de datos de la que se consultarán a cada instante los últimos datos insertados y así monitorizar el estado del proceso quirúrgico.

Antes de iniciar la detección de anomalías deben seguirse los siguientes pasos:

1. El primer paso para poner en marcha el sistema es tener montado y conectado el circuito al ordenador que se encargará de enviar los datos a la base de datos. Para que funcionen correctamente los programas creados, los sensores deben conectarse como se indica en las figuras 21 y 22.

   Posteriormente, se conectan cada una a un puerto USB del equipo que enviará los datos.

2. El segundo paso sería compilar en el Arduino IDE los ficheros arduino_uno.ino y arduino_mega.ino para posteriormente enviarlos a su respectiva placa. Se comprueba que funcionan correctamente abriendo el monitor serie desde Arduino IDE y viendo que los sensores están enviando datos.

3. Para enviar las medidas recogidas por los sensores a la base de datos es necesario que esta esté abierta. Para poner en marcha la base de datos debemos abrir una consola de comandos en "MongoDB\Server\4.2\bin" y añadir la instrucción que se muestra a continuación para iniciar mongo con el archivo de configuración adecuado.

```
$ mongod --config mongod.cfg
```

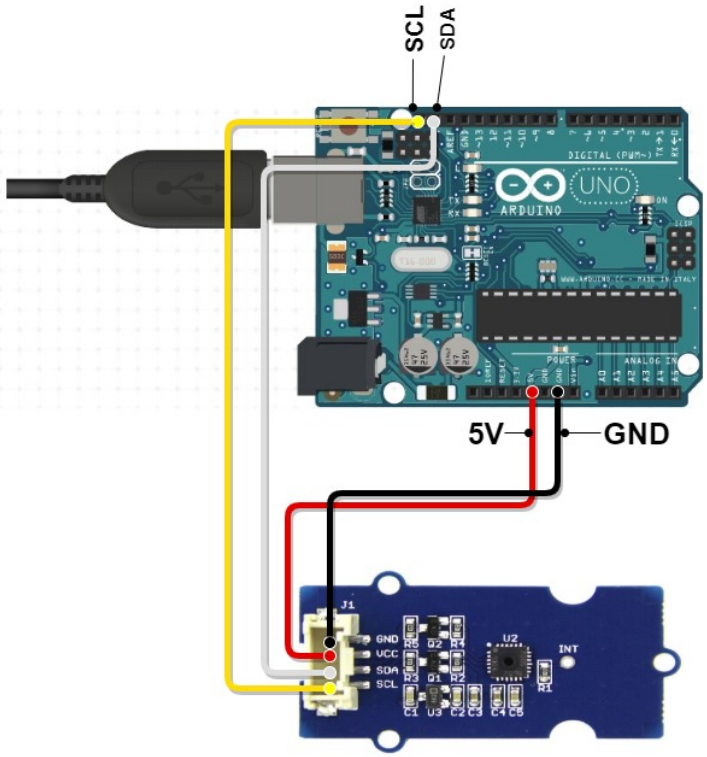


Figura 21: Circuito Arduino Uno

4. Iniciar el envío de datos a MongoDB, en el caso de las medidas capturadas por los sensores se enviarán desde el equipo que tenga conectadas las placas Arduino. Para realizar el envío de las medidas se debe ejecutar el archivo "send_measurements.py". Para que también sea comprobado el origen de los paquetes de datos debemos ejecutar en el equipo en el cual se encuentre la base de datos el archivo "packet_data_train.py".

5. Una vez se estén enviando los datos a MongoDB, pondremos en marcha el algoritmo de detección de anomalías ejecutando el archivo "predict.py".

6. Pulsar el botón "Iniciar detección de anomalías" para ver lo que ocurre en la sala.

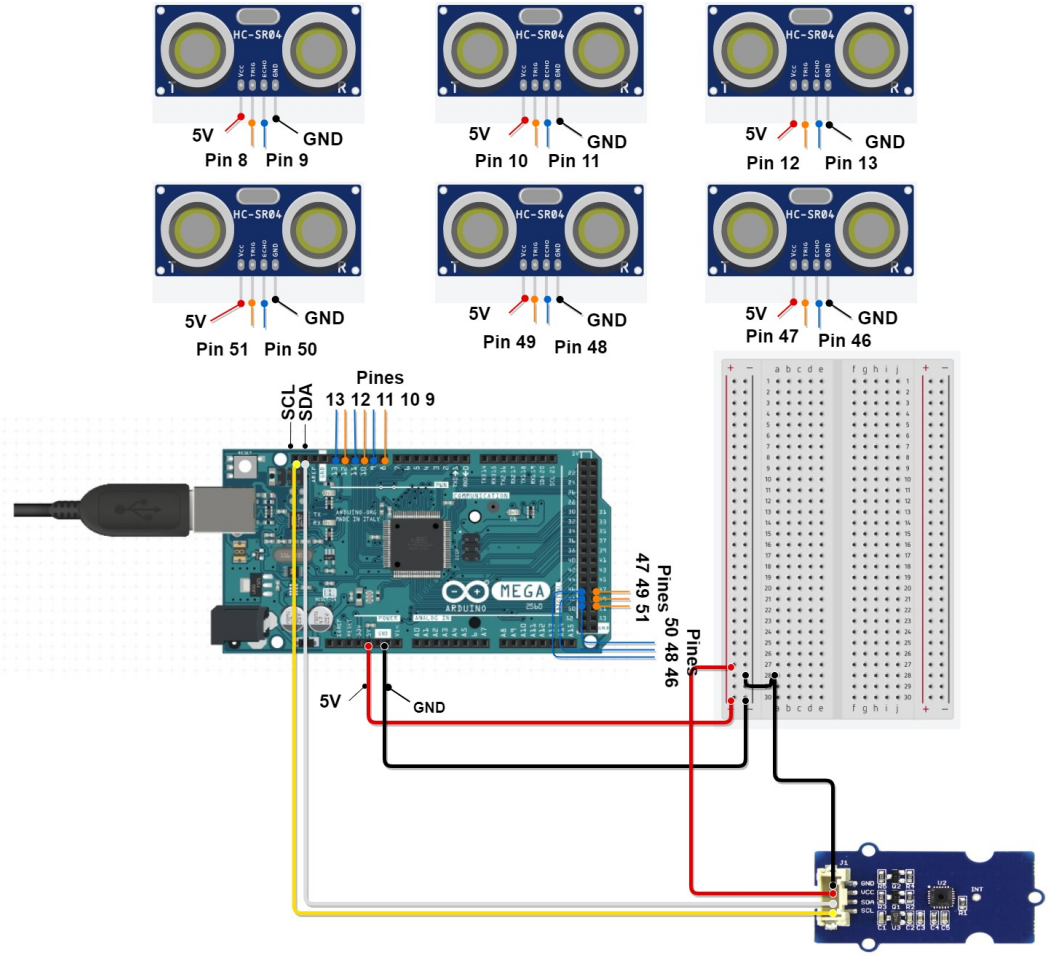


Figura 22: Circuito Arduino Mega

B. Manual de Instalación

En este anexo se exponen los programas y aplicaciones necesarios, así como los pasos que se deben seguir para la instalación del sistema:

Arduino

Circuito

El esquema del circuito para el sistema debe ser el que se muestra en las Figuras 21 y 22. Ambas placas deben estar enchufadas al equipo que envíe la información a la base de datos.

Arduino IDE

Para enviar los programas .ino a las placas es necesario tener instalado el entorno de Arduino[1] o utilizar el editor online[2].

Base de datos Mongo

En primer lugar es necesario descargar e instalar el software de Mongo Community Edition[3] en la máquina en la que vaya a ser alojada la base de datos. A continuación, existen dos posibilidades para poner en marcha la base de datos:

Opción 1: Importar Base de Datos

[1]El software puede descargarse de la página oficial: `https://www.arduino.cc/en/main/software`

[2]Página de acceso al editor online: `https://create.arduino.cc/`

[3]El software está disponible en: `https://docs.mongodb.com/manual/administration/install-community/`

```
mongorestore --Host = <ip> --port = 27017 -d <direccion_archivo>
```

Opción 2: Crear Base de Datos

El nombre de la base de datos a crear será "cps", para crear los usuarios y colecciones debemos indicar la base de datos con la siguiente instrucción:

```
use cps
```

Crear Colecciones

Para que todo funcione correctamente se debe haber las siguientes colecciones en la base de datos:

1. **sensors_data**
2. **sensors_data_train**
3. **packet_data**
4. **packet_data_train**
5. **predict_log**

La instrucción con la que se crean las colecciones en Mongo es la siguiente:

```
db.createCollection("nombre_coleccion")
```

Crear Usuarios

La instrucción con la que se crean los usuarios con los permisos correspondientes mediante la siguiente instrucción:

```
db.createUser(
...
... {
... user: "user_database",
... pwd: passwordPrompt(),
... roles: [
... { role : "readWrite", db: "cps"}
... ]
... }
... )
* En roles incluir una lista con los roles requeridos
```

Los usuarios con sus permisos son:

Usuario	**Base de Datos**	**Permisos**
admin	admin	"userAdminAnyDatabase"
arduino	cps	"readWrite": cps
machine-learning	cps	"readWrite": cps
unity	cps	"read": cps
sniff	cps	"readWrite": cps

Tabla 5: Usuarios y permisos

Python

Instalar la versión de Python 3.6[4] en adelante. Además de las librerías: Pandas, Keras, Pymongo, scikit-learn y scapy.

[4]Software disponible en: `https://www.python.org/downloads/`

Aplicación Unity

Para la instalación de la aplicación Unity que contiene la interfaz gráfica de control únicamente hay que ejecutar el archivo ejecutable disponible para Windows 10, MacOS y Linux. En el caso de windows existe un instalador.

C. Entrevista con el Doctor

Entrevista a un doctor del Hospital de Alta Resolución de Benalmádena.

Preguntas generales

- ¿Cómo es la distribución de una sala quirúrgica básica (mesa de operaciones, pantallas, ecg, luces, puertas, ventanas)? ¿Cuántas salidas necesita tener el quirófano? ¿Y ventanas?

  Los quirófanos suelen ser cuadrados y tener una entrada amplia para que entren camillas. Además, cuentan con otro acceso, al área de sucio, zona en a la que se lleva el instrumental empleado en la operación. El instrumental limpio entra listo para ser utilizado por la puerta principal y sale por la puerta de sucio, es lo que se conoce como circulación de limpio y de sucio.

  Camilla en posición central y lámpara centrada encima de la camilla. Torre de gases al lado de la pared con brazo articulado. Aparatos de anestesia. Mesa instrumental. Sería bueno tener mesa para sentarse y escribir. La sala tiene que tener fácil desmontaje para limpiar rápidamente.

- Durante una intervención, ¿qué condiciones ambientales deben estar controladas? (nivel de temperatura, nivel de humedad, luz, etc.)

  El quirófano debe contar con unas buenas condiciones de luminosidad. Presión por encima de 5 minimo – entre 5 y 10. Presión el quirófano superior a la de fuera. El aire siempre sale nunca entra. Lo mas importante: humedad, presión y temperatura.

  Todas estas condiciones vienen detalladas en el manual: Bloque quirúrgico.

- ¿Qué constantes vitales del paciente deben controlarse? En cuanto al pulso cardíaco, cómo se mide, medidas normales, medidas de alerta, etc.

  Pulso simetría, electrocardiograma..

- ¿Qué tipo de robots quirúrgicos se están utilizando actualmente? (Qué tipo de operaciones se pueden realizar con dichos robots)

  Todos los robots que se emplean actualmente en quirófano son robots esclavos, lo que quiere decir que no tienen autonomía. Son los cirujanos los que ordenan los movimientos que deben realizar. Algunos de los que se están empleando actualmente son: Robot: Zeus (menos completo que el DaVinci). Zeus fue uno de los originales. DaVinci se controloa con la frente y pedales. Zeus es un manipulador de camaras. DaVinci en civil, virgen del rocio y sanis. Cada repuesto entre 5000 y 6000€y mantenimiento elevado. DaVinci es robot esclavo.

Preguntas específicas

- ¿En qué tipo de intervenciones se suele usar (el robot previamente dicho)? De esas intervenciones, ¿cuál es la que considera menos compleja?

  Una intervención simple sería la extracción de un lipoma. Aunque actualmente no son muy utilizados debido al alto coste del mantenimiento y de los recambios de piezas.

- ¿Cuáles son las fases de dicha intervención en condiciones normales?

  Intervención sencilla: Paciente llega al transfer, entra al quirófano, identificación correcta del paciente. Se le identifica con la pulsera (Comprobar nombre y fecha nacimiento), alergias, si es de intubación difícil, confirmar procedimientos (de que se va a operar ej: hernia, de que lado). Luego se monitoriza, electrocardiografía, después anestesia y se entuba una vez dormido. Se controla la respiración. A continuación, pintar con antiséptico la zona de la incisión. Se colocan paños quirúrgicos y se prepara la instrumentación. Se toma el bisturí eléctrico y luego ya se procede a la operación. Cuando se termina se despierta al paciente y se lleva a la sala de recuperación y allí se monitoriza al paciente igual que durante la operación.

- ¿Qué tipo de "anomalías"(condiciones no normales) pueden ocurrir en una sala

quirúrgica y que pueden afectar a la intervención?

La anomalía más común en un lipoma es que se baje la tensión. Esto se controla por la pulxiosimetría. Cuando se detecta, se para la operación y se incorpora el paciente. Además, con respecto a los factores ambientales, si se detectan descompensaciones en estos valores y no se pueden recuperar la intervención se detiene.

- Independientemente del Robot, cuando se operan a personas, qué tipo de tecnologías o sistemas informáticos se aplican para controlar estados de la infraestructura y del paciente? ¿o no existen?

  Se controla, la humedad, presión y temperatura de la sala. Si la temperatura sube o baja de los valores normales, al igual que ocurre con la presión y temperatura se envía un aviso al sistema central.

- ¿Qué protocolos de emergencia existen cuando ocurre alguna anomalía (alarmas sonoras, visuales, luces de emergencia, etc.)?

  Suelen ser alarmas sonoras. Y luces de emergencia cuando se va la luz. En los robots suele haber un botón rojo para bloquear el sistema y poder manipular.

- Tipo de seguridad o política de seguridad (a nivel de infraestructura/sala de operaciones) se establece para proteger las salas de cirugía.

  Para poder tener acceso a las salas de cirugía hay que disponer de una acreditación. La diferente instrumentación que es usada durante la intervención quirúrgica es controlada mediante códigos QR por lo que se conoce que material ha entrado en una sala.

  Por otro lado, para el suministro de medicamentos a los pacientes, solo se puede tener acceso a ellos mediante identificación del personal sanitario.

- Tipo de seguridad o política de seguridad (a nivel de información) se establece para proteger el acceso a las salas.

  Los equipos informáticos que se encuentran en las salas solamente son accesibles mediante usuario y contraseña. Además, para poder llegar hasta la sala existen otras

medidas de seguridad que hacen más complejo el acceso no autorizado a estos equipos.

- ¿Cómo se gestiona la información que se controla de los pacientes? ¿Existe un sistema centralizado que recopila dicha información?

  Sí, la información de los pacientes se encuentra centralizada

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA